AÑO III - NUM. 73

MICROHOBBY SEMANAL

REVISTA INDEPENDIENTE PARA USUARIOS DE ORDENADORES SINCLAIR

AÑO III - N.º 73

135 PTS.
Canarias 140 ptas.

INFORME

TODO SOBRE LA PROPIEDAD INTELECTUAL DE LOS PROGRAMAS

NUEVO

MOVIE: EL DETECTIVE MARLOW SE ENFRENTA AL MUNDO DEL HAMPA

HARDWARE

COMO FUNCIONA EL MODULADOR DE VIDEO EN EL SPECTRUM

PROFESOR PARTICULAR

DESPEJAR INCOGNITAS



INCREÍBLE MAPA FOTOGRÁFICO A TODO COLOR

HOBBY PRESS

Director Editorial
José I. Gómez-Centurión
Director Ejecutivo
Domingo Gómez
Asesor Editorial
Gabriel Nieto
Redactora Jefe
Africa Pérez Tolosa
Diseño
Rosa María Capitel
Redacción
Amalio Gómez, Pedro Pérez, Jesús Alonso
Secretaría Redacción
Carmen Santamaría
Colaboradores
Primitivo de Francisco, Rafael Prades, Miguel Sepúlveda, Sergio Martínez y J. M. Lazo
Corresponsal en Londres
Alan Heap
Fotografía
Carlos Candel
Portada
José María Ponce
Dibujos
Teo Mójica, F. L. Frontán, J. M. López Moreno, J. Igual, J. A. Calvo, Lóriga, J. Olivares

Edita
HOBBY PRESS, S. A.
Presidente
María Andrino
Consejero Delegado
José I. Gómez-Centurión
Jefe de Producción
Carlos Peropadre
Jefe de Publicidad
Concha Gutiérrez
Publicidad Barcelona
José Galán Cortés
Tels.: 303 10 22 - 313 71 76
Secretaría de Dirección
Marisa Cogorro
Suscripciones
M.ª Rosa González
M.ª del Mar Calzada
Redacción, Administración y Publicidad
La Granja, 39
Polígono Industrial de Alcobendas
Tel.: 654 32 11
Telex: 49480 HOPR
Dto. Circulación
Paulino Blanco
Distribución
Coedis, S. A. Valencia, 245
Barcelona
Imprime
Rotedic, S. A. Ctra. de Irún, km 12,450 (MADRID)
Fotocomposición
Novocomp, S.A.
Nicolás Morales, 38-40
Fotomecánica
Grof
Ezequiel Solana, 16
Depósito Legal
M-36.598-1984

Representante para Argentina, Chile, Uruguay y Paraguay, Cia. Americana de Ediciones, S.R.L. Sud América 1.532. Tel.: 21 24 64. 1209 BUENOS AIRES (Argentina)

Solicitado control OJD

MICROHOBBY ESTA SEMANA

AÑO III. N.º 73. 8 al 14 de abril de 1986.
135 ptas. Canarias, Ceuta y Melilla: 130 ptas.
Sobretasa aérea para Canarias: 10 ptas.

4 **MICROPANORAMA.** Publicamos un amplio informe sobre el proyecto de ley de la propiedad intelectual.

7 **TRUCOS.**

8 **PROGRAMAS MICROHOBBY.** Mark.

10 **NUEVO.** «Movie», «Macadam Bumper», «Robin of Sherlock» y «Show Jumping».

14 **TRAS LA SOMBRA DEL UNICORNIO.**

15 **CODIGO MAQUINA.**

23 **PROFESOR PARTICULAR.** Un programa para «Despejar» la «x» de cualquier expresión.

24 **HARDWARE.** El modulador de vídeo.

28 **ESTRATEGIA.** Batcode (III).

30 **LOS JUSTICIEROS DEL SOFTWARE.**

31 **MICROMANIA.**

32 **CONSULTORIO.**

34 **OCASION.**





El modulador de vídeo. (Pág. 24).

PREMIADOS HOBBY-SUERTE

ALVARO PEREZ MULAS. C/ Arco. (SALAMANCA).
Suscripción a Microhobby Semanal por un año (4.ª Cat.).

M.ª LUISA FUENTES BLANCO. C/ Gericó, 3, Bjo. (MADRID).
Un Joystick con su Interface (3.º Cat.).

ANGEL FERNANDEZ RUIZ. C/ Sin Pecado, 6, 8.º D. (SEVILLA).
Una cinta de programas (5.ª Cat).

PILAR GAYOSO RODRIGUEZ. Avda. Gral. Primo de Rivera, 5, 2.º D. (LOGROÑO).
Suscripción a Microhobby Semanal por un año (4.ª Cat).

FCO. JAVIER ORTEGA MAUTO. C/ Penélope, 6, 2.º C. Barajas. (MADRID).
Una cinta de programas (5.ª Cat.)

ISABEL CASADO CALDERILLA. C/ Zigia, 14. (MADRID).
Una impresora Seikosha (2.ª Cat.).

JORDI VENTURA BALLUS. C/Pins, 28. La Garriga. (BARCELONA).
Una cinta de programas (5.ª Cat.)

ANTONIO GIL POZOS. C/ Ernest Hemingway, 6. Dos Hermanas. (SEVILLA).
Una cinta de programas (5.ª Cat.)

ANTONIO MARTIN SEVILLA. C/ Angli, 4, Bis. (BARCELONA).
Una cinta de programas (5.ª Cat.)

TOMAS RODRIGUEZ SUAREZ. C/ Barcelona, 3, 3.º D. (LAS PALMAS DE G.C.).
Una cinta de programas (5.ª Cat.)

MANUEL GONZALEZ LEAL. C/ Naos Pie, 10, 2.º D. (SEVILLA).
Una cinta de programas (5.ª Cat.)

Los derechos de propiedad intelectual de programas

RECONOCIDOS, POR FIN, EN EL B.O.E

En semanas anteriores anticipábamos la noticia de la creación, en el Congreso de los Diputados, de un Proyecto de Ley de la Propiedad Intelectual. En este número os presentamos un informe más detallado de todos aquellos puntos que por su relación con la informática, pueden despertar un mayor interés.

En el Boletín Oficial de las Cortes Generales correspondientes al 1 de febrero de 1986, aparece un Proyecto de Ley de Propiedad Intelectual que, por primera vez, contempla un título dedicado a los programas para ordenador. Este proyecto surge de la necesidad de renovar la anticuada Ley vigente, promulgada en el año 1879, y tiene como principales objetivos delimitar los derechos y obligaciones de los autores y explotadores de obras tecnológicamente avanzadas, como son las audiovisuales o los programas de ordenador.

Evidentemente, el Proyecto incluye en su texto un gran número de artículos y capítulos dedicados a todos los temas relacionados con la creación intelectual o artística, por lo que nos limitarenos a tratar sólo aquéllos que tengan que ver, de una u otra forma, con los derechos y obligaciones de los creadores o usuarios de los programas para ordenador.

En primer lugar cabría destacar uno de los párrafos que se incluyen dentro del apartado de la Exposición de motivos, el cual afirma que:

«el reconocimiento de los derechos de propiedad intelectual no está sujeto a requisitos formales de ningún tipo, la Ley faculta a los titulares de los mismos para que, como medida especial de protección y salvaguarda, procedan a su inscripción en el Registro de la Propiedad Intelectual.»

del cual se desprende que es errónea la creencia generalizada de que para adquirir los derechos sobre cualquier creación es necesario el haberla registrado previamente.

Pero pasemos a exponer los títulos y artículos más destacados:

LIBRO I

DERECHOS DE AUTOR

TITULO I

DISPOSICIONES GENERALES

Artículo 1

El autor de una obra literaria, científica o artística adquiere la propiedad intelectual sobre ella por el solo hecho de su creación.

CAPITULO II

Objetivo

Artículo 10

1. Son objeto de propiedad intelectual todas las creaciones originales literarias, científicas y artísticas expresadas por cualquier medio o soporte, tangible o intangible, actualmente conocido o que se invente en el futuro.

Como se puede observar en este artículo, comprende cualquier tipo de creación de una forma muy generalizada, pero posteriormente se ofrece una larga lista de posibles creaciones, entre las que se incluyen los programas de ordenador.

En el Capítulo III, Sección Primera, dedicado al Derecho Moral de los autores, cabría destacar los siguientes puntos del artículo 14:

Artículo 14

El autor goza de las siguientes facultades irrenunciables e inalienables.

1. Decidir si su obra ha de ser divulgada y en qué forma.

4. Exigir el respeto a la integridad de la obra e impedir cualquier deformación, alteración o atentado contra ella que suponga perjuicio a sus legítimos intereses o menoscabo a su reputación.

6. Retirar la obra del comercio, por cambios de sus convicciones intelectuales o morales, previa indemnización de daños y perjuicios a los titulares de derechos de explotación.

Mientras que, por su parte, la Sección Segunda dedicada a los derechos de explotación, expone en su artículo 17, lo siguiente:

Artículo 17

Corresponde al autor el derecho exclusivo a la explotación de su obra en cualquier forma y, en especial, mediante su reproducción, distribución, comunicación pública y transformación, que no podrán ser realizadas sin su autorización.

el cual es bastante explícito por sí mismo y otorga todos los derechos en cualquiera de sus formas, al creador de la obra.

En el título relativo a la transmisión de los derechos por parte del autor o otra persona o personas, cabe destacar los siguientes artículos que tratan sobre las remuneraciones que los creadores deben percibir.

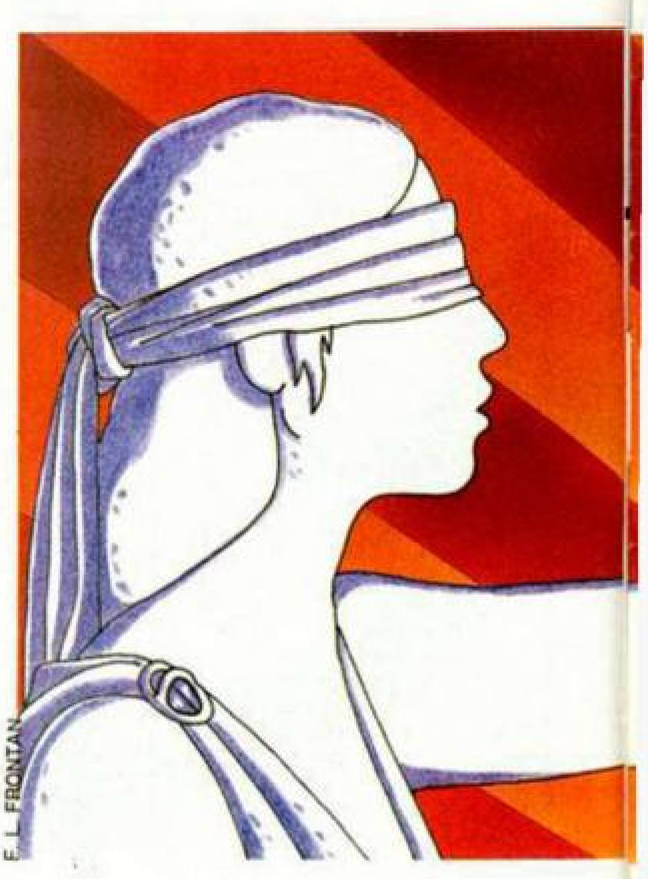
Artículo 46

1. La cesión otorgada por el autor a título oneroso le confiere una participación proporcional en los ingresos de la explotación, en la cuantía convenida con el cesionario.

En cuanto a los derechos del usuario, este capítulo también es bastante explícito:

Artículo 56

1. El adquirente de la propiedad del soporte a que se haya incorporado la obra no tendrá por este solo título nin-



gún derecho de explotación sobre esta última.

Y pasando ya directamente el título VII, dedicado exclusivamente a los programas de ordenador, destacamos tres de los seis artículos que contiene y que se refieren a la propia definición de programa y a otros temas de su interés, no sólo para el autor sino también para el usuario de los mismos.

Artículo 95

1. A los efectos de la presente Ley se entenderá por programa de ordenador toda secuencia de instrucciones o declaraciones destinadas a ser utilizadas, directa o indirectamente, en un sistema informático para obtener un resultado determinado, cualquiera que fuere su forma de expresión y fijación.

4. La protección establecida en la presente Ley se extiende a cualesquiera versiones sucesivas del programa, así como a los programas derivados.

Por su parte, el punto 2 del artículo 98, especifica claramente que:

2. La reproducción del programa incluso para uso personal exigirá la autorización del titular del derecho de explotación con excepción de la copia de seguridad.

lo cual unido al punto 4

4. No constituye transformación a los efectos previstos en el artículo 21, la adaptación de un programa realizada por el usuario para la utilización exclusiva por el mismo.

deja bastante claro que el usuario de un programa, tiene el derecho legal de poder realizar una copia de seguridad o bien llevar a cabo en él las modificaciones oportunas para adaptar el programa a sus necesidades específicas, siempre y cuando, eso sí, lo haga con fines de uso personal.

Esta ley se contrapone notablemente a la tendencia habitual de las casas de software de proteger al máximo sus copias, dificultando enormemente a los usuarios la posibilidad de ejercer este derecho.

Sin embargo, estas medidas de protección, también tienen su lado de lógica debido al turbio asunto de las copias ilegales, tema éste también contemplado en el Proyecto de Ley.

Artículo 123

El titular de los derechos reconocidos en esta Ley, sin perjuicio de otras acciones que le correspondan, podrá instar el cese de la actividad ilícita del infractor y la de indemnización de los daños materiales y morales causados.

Artículo 124

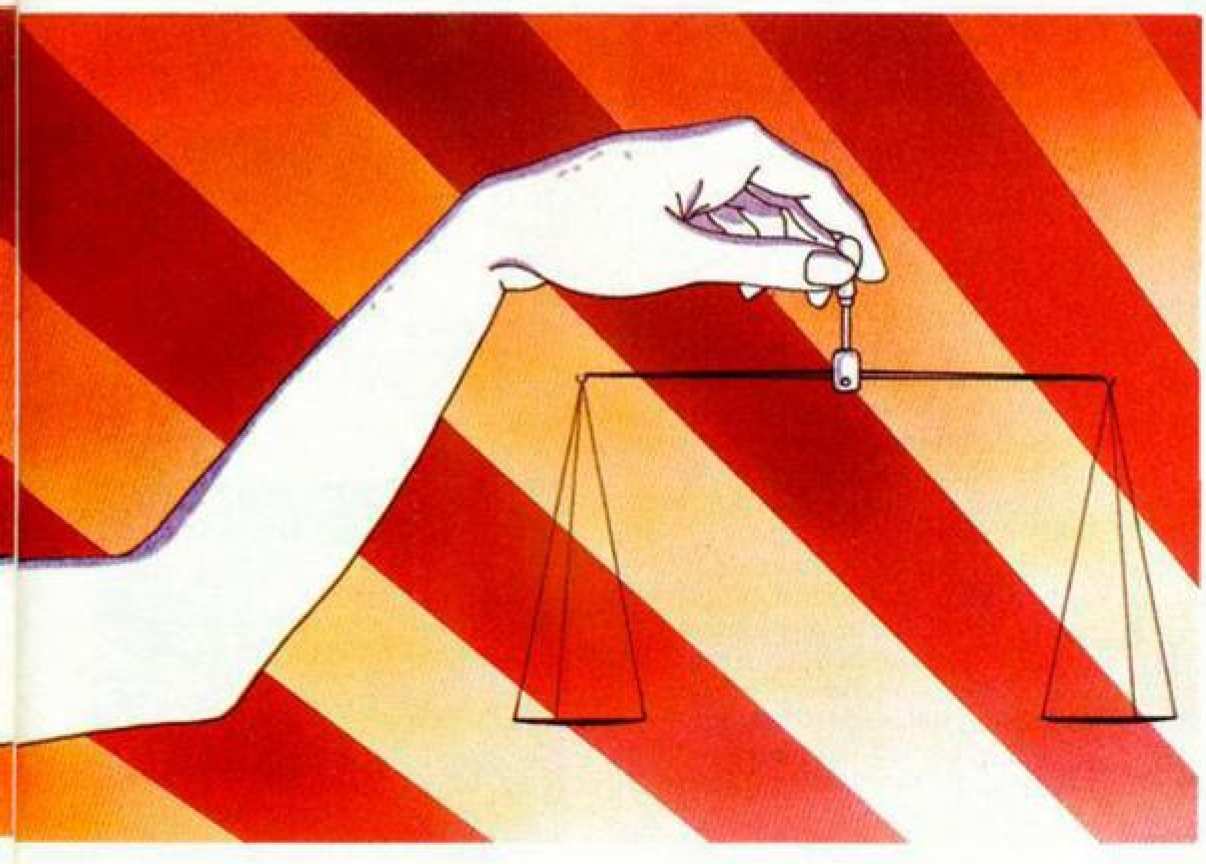
1. El cese de la actividad ilícita podrá comprender:

a) La suspensión de la explotación infractora.

b) La prohibición al infractor de reanudarla.

c) La retirada del comercio de los ejemplares ilícitos y su destrucción.

d) La inutilización de los moldes, planchas, matrices, negativos y demás elementos destinados exclusivamente a la reproducción de ejemplares ilícitos y, en caso necesario, la destrucción de tales instrumentos.



Del mismo modo, en la Sección III titulada: «De los delitos contra la propiedad intelectual e industrial», se deduce la gran dureza con la que pueden ser tratadas aquellas personas que infrinjan estas leyes, y en los siguientes artículos se señalan algunas de las penas que pueden sufrir estos infractores:

Artículo 534 bis, a)

Será castigado con la pena de multa 30.000 a 600.000 pesetas quien intencionadamente reprodujere, distribuyere o comunicare públicamente, en todo o en parte, una obra literaria, científica o artística o su transformación o una interpretación o ejecución artística fijada en cualquier tipo de soporte o cumunicada a través de cualquier medio, sin la autorización de los titulares de los correspondientes derechos de propiedad intelectual o de sus cesionarios.

Artículo 534 bis, b)

Será castigado con la pena de arresto mayor o multa de 50.000 a 1.500.000 pesetas quien realizare cualquiera de las conductas tipificadas en el artículo anterior, concurriendo alguna de las siguientes circunstancias:

a) Obrar con ánimo de lucro.

b) Infringir el derecho de divulgación del autor.

c) Usurpar el derecho de la condición de autor sobre una obra o el nombre de un artista, en una interpretación o ejecución.

d) Modificar sustancialmente la integridad de la obra sin autorización del autor.

Con estos artículos se puede resumir a grandes rasgos el contenido de este Proyecto de Ley, el cual viene a paliar la necesidad absoluta de una legislación que amparase a los temas derivados de software, pues no cabe duda que la creciente importancia de la informática en nuestra sociedad obliga a la promulgación de una Ley que la contemple en todos sus aspectos.

Para finalizar, simplemente señalar que en una reciente rueda de prensa, convocada por ANEXO, se manifestó el total apoyo y acuerdo con este proyecto y es de esperar que esta Ley reciba idéntica aceptación por parte de los programadores y usuarios en general.

MICROPANORAMA

AQUI LONDRES



Las últimas noticias sobre el nuevo microordenador portátil **'Pandora'** de Sinclair, son de que no estará disponible hasta el próximo año. Posiblemente se pondrán unos modelos de desarrollo a disposición de la industria y la prensa dentro de algunos meses, pero Sinclair no lanzará la nueva máquina al público hasta 1987.

El juego mejor vendido durante el pasado año fue **'The Way of the Exploding Fist'** de Melbourne House, que también fue n.º 1 para Amstrad durante el año 1985.

'Softaid', la cinta de recopilación, se llevó el premio para Spectrum y Commodore 64, mientras que 'Exploding Fist' quedó segundo en ambos casos.

Un nuevo monitor en color con alta resolución, será lanzado por **Chromographic**. Este monitor ha sido diseñado especialmente para el Spectrum 128K, a pesar de que no posee, por desgracia, salida de sonido.

Bug-Byte ha decidido lanzar al mercado una gama de software barato que, en principio, sólo valdría para Commodore 64 y 128. Esta serie llevará el nombre de 'Load and Go' y consiste en una serie de títulos escritos en los EEUU.

La serie incluye un procesador de textos, una agenda, una base de datos, una hoja de cálculo y dos programas de contabilidad. El precio en el Reino Unido será de 8 libras.

De nuestro corresponsal en Londres
ALAN HEAP

JUEGOS DE INVIERNO EN PRIMAVERA



Un nuevo juego de simulación deportiva acaba de hacer su aparición en el mercado español. Se trata de Winter Games, programa éste creado por Epyx y distribuido en nuestro país por Compulogical.

Winter Games es una maravillosa recopilación de siete pruebas diferentes relacionadas con el hielo y la nieve, de las cuales se puede decir que todas mantienen un nivel de calidad muy elevado, tanto en sus aspectos gráficos como de divertimento.

Los deportes que configuran este programa son de la más variada naturaleza, pero todos ellos resultan igualmente adictivos: Salto de Sky, Sky acrobático, Patinaje de Velocidad, Patinaje artístico, Bobsled, Biathlon y Patinaje estilo libre.

La versión de este juego para Commodore lleva ya algún tiempo en el mercado y si bien éste supera en muchos detalles al realizado para Spectrum, no es por otro motivo que por la mayor capacidad de memoria de dicho ordenador, aunque en lo referente a la adicción no existe ninguna diferencia entre ambos.

Sin duda alguna, y a pesar de lo ya avanzado de la primavera, en los próximos meses todos nos iremos a hacer deporte a la nieve.

Celebrado en Extremadura

I CONCURSO DE PROGRAMAS DIDACTICOS PARA ORDENADOR

Con motivo de la VII Semana de Extremadura en la Escuela, se convocó el I Concurso de Programas Didácticos para Ordenadores con el patrocinio del Instituto de Ciencias de la Educación de la Universidad de Extremadura.

El resultado del mismo ha sido una cinta en la que se incluyen los tres programas ganadores, todos ellos relacionados muy directamente con la educación a través de la informática. Estos programas son: *Extremadura,* el cual contiene tres bloques dedicados a dar información a los alumnos de la segunda Etapa de EGB de los temas relacionados con la geografía de dicha región autonómica; *Máquina,* un juego destinado a enseñar a los más pequeños las operaciones elementales con los números, y por último, *Soto,* formado por un archivo en el que se encuentran los temas más importantes tanto geográficos como históricos de Extremadura.

Estas actividades no tienen ningún fin lucrativo, por lo que el precio de las cintas es realmente reducido (300 pts).

La iniciativa es interesante y puede ser tomada como ejemplo por otras comunidades autonómicas que deseen potenciar las relaciones entre la informática y la educación en sus escuelas.



TRUCOS

CUESTION DE CABECERAS

La finalidad de esta pequeña rutina en Código Máquina que nos ha enviado Juan Tortosa, y que está controlada desde el Basic, no es otra que la de crear o cargar un bloque de bytes sin cabecera.

Para hacer todo ello, es indispensable dar la dirección de comienzo y la longitud.

Un consejo: apuntar esos datos al salvar un bloque para, más tarde, poderlo cargar.

La rutina es reubicable y sólo ocupa 27 bytes.

```
9997 POKE 23658,8: INPUT "DIRECC
ION COMIENZO: ";C: INPUT "LONGIT
UD: ";L: INPUT "DIRECCION DE LA
RUTINA: ";D: LET A2=INT (C/256):
 LET A1=C-(256*A2): LET B2=INT (
L/256): LET B1=L-(256*B2)
9998 FOR F=D TO D+26: READ P: PO
KE F,P: NEXT F: DATA 221,33,A1,A
2,17,B1,B2,62,255,205,194,4,201,
62,255,17,B1,B2,221,33,A1,A2,55,
205,86,5,201: INPUT "CARGAR O SA
LVAR (C/S): ";A$: IF A$="C" THEN
 RANDOMIZE USR (D+13): STOP
9999 PRINT "PULSA TECLA": PAUSE
0: RANDOMIZE USR D: STOP
```

EFECTO DE «PAPER»

La rutina que nos ha mandado desde Huelva Luciano Bueno crea un efecto del «paper» de la pantalla, combinando colores para crear sucesivos rectángulos, que van haciéndose cada vez más pequeños. Todo ello, se mezcla oon un curioso efecto sonoro.

Este truco es muy útil para conseguir presentaciones vistosas de programas, tanto en Basic como en Código Máquina, pues la rutina es reubicable en cualquier dirección de memoria.

```
 10 FOR n=50000 TO 60123
 20 READ a: POKE n,a: NEXT n
 30 DATA 175,50,4,91,62,31,50,0
,91,62,23,50,1,91,33,0,88,34,2,9
1,6,12,197,42,2,91,58,0,91,71,58
,4,91,119,35,16,252,58
 40 DATA 1,91,71,58,4,91,17,32,
0,119,25,16,252,42,2,91,58,1,91,
71,58,4,91,17,32,0,119,25,16,252
,58,0,91,71,58,4,91,119
 50 DATA 35,16,252,119,42,2,91,
1,33,0,9,34,2,91,33,0,91,53,53,3
5,53,53,38,1,58,4,91,111,17,10,0
,205,181,3,193,16,165,58,4,91,25
4,56,200,198,8,24,134,201
 60 RANDOMIZE USR 50000
```

TRUCOS FANTASMAS

Un asiduo lector, Juan José Moya, nos ha mandado un truco que según sus palabras, servirá para completar los que salieron publicados en el n.º 64 de la revista con el título «Sonido y Borders».

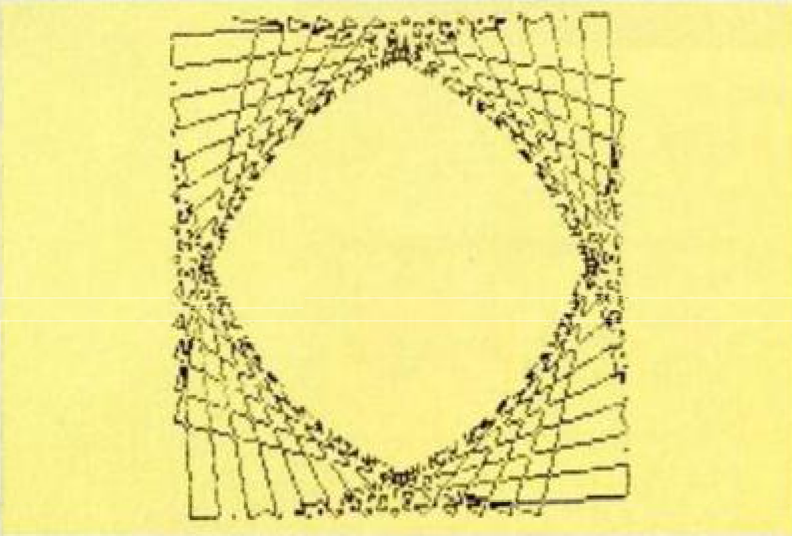
El suyo hace todo lo contrario a lo que hemos publicado hasta el momento (léase, ampliar juegos de caracteres) y consiste en reducirlos, pudiéndose escribir en 64 columnas, si bien este procedimento hace que algunos caracteres que no salgan muy legibles.




```
  2 REM  64 columnas
  3 INPUT "Posicion vertical
(ENTER) ";PA,"Posicion horizonta
l (ENTER) ";PO
  4 INPUT LINE A$: PRINT INK 7;
AT 0,0;A$
  5 LET H=PO: LET V=PA
 10 FOR F=168 TO 175
 20 FOR G=0 TO LEN A$*8-1
 30 IF POINT (G,F)=1 THEN PLOT
INT H,V
 31 LET H=H+.5
 40 NEXT G: LET H=PO: LET V=V+1:
NEXT F
```

PARA DIBUJAR

Si quieres conseguir este dibujo para adornar tu pantalla, teclea la rutina que ha hecho Mikel Elósegui.



```
  5 OVER 1
 10 FOR f=0 TO 155
 20 PLOT 50,f
 30 DRAW f,155-f
 40 DRAW 155-f,-f
 50 DRAW -f,-155+f
 60 DRAW -155+f,f
 70 NEXT f
 80 GO TO 10
```

Si deseas que la figura sea cuadrada, cambia la línea 10, por esta

```
10 FOR f=0 TO 155 STEP 10
```

Y si lo que pretendes es que el dibujo, uno u otro, gire, añade la línea 65.

```
65 CLS
```

MARK

José Ignacio VICARIO

Spectrum 48 K

Mark es un intrépido aventurero con una vida plagada de odiseas y hazañas que se van haciendo leyenda por donde pasa. En esta ocasión, se dispone a emprender otra que le llevará al «Gran Templo.»

Situado en medio de una inmensidad selvática, se encuentra este «Gran Templo» (así conocido por los nativos) en cuyo interior dicen que existe un magnífico tesoro custodiado por un sinfín de peligros (estatuas que nos desintegran, flechas, esqueletos...) Nuestra misión será, como únicos protagonistas de la aventura, la de recoger las 20 llaves que se encuentran esparcidas por las distintas cavernas que componen el templo. Para ello contamos con cuatro vidas, un tiempo limitado y un poco de suerte.

Las teclas del movimiento son las siguientes:

O, para ir a la izquierda.

P, para hacerlo hacia la derecha.

Q, para saltar a derecha o izquierda, según mire Mark.

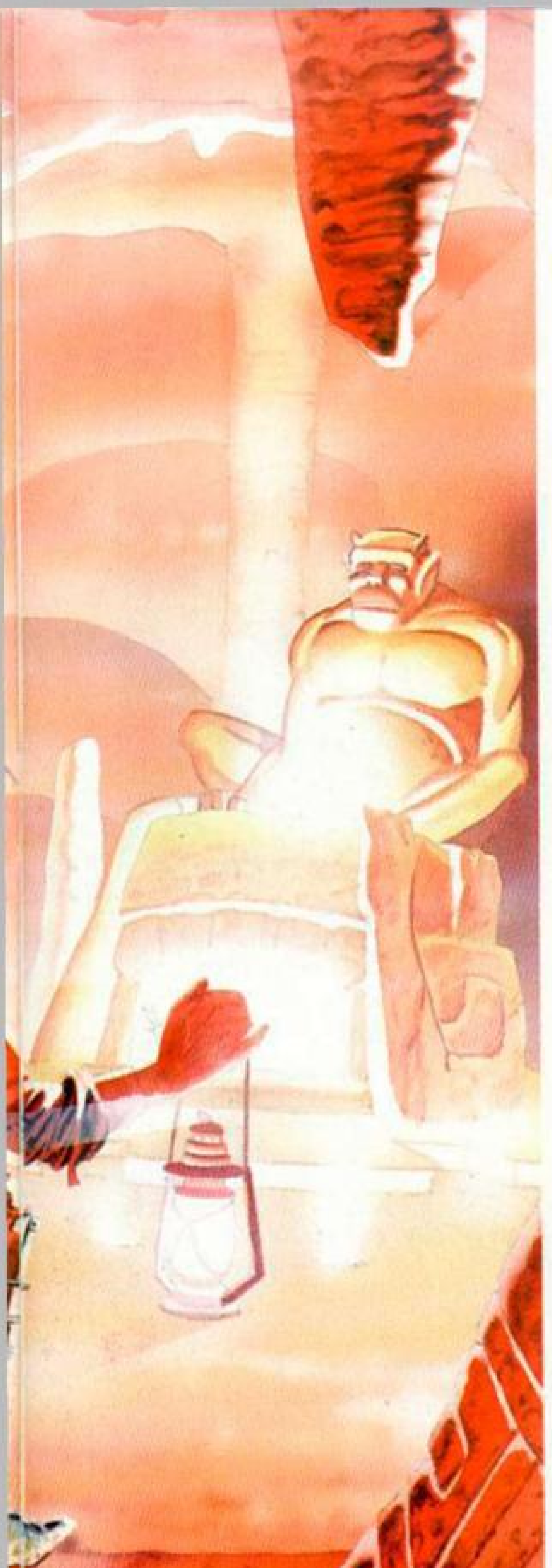
U, para subir.

J, para bajar.





```
   1 PAPER 1: BORDER 0: CLS : CLEAR 65199: LET pus=0: LET du=0: LET n$="A A A A ": LET m$="D D D D ": GO SUB 9920: GO SUB 9950: GO SUB 9980: GO SUB 8000: LET b$="A": LET c$="D": LET pis=0: LET i$="bbbbbbbbbbbbbbbbbbbbbbb": LET [redacted]": LET t$="  E  E  E ": LET c=4: LET d=1
   2 RESTORE 9700+pis: GO SUB 7500: CLS : PRINT INK 4;AT 0,0;r$;s$;t$: GO SUB 7000: GO SUB 9000: GO SUB 4000+pis: GO SUB 2010
   4 IF i$((pis/10)+1)="b" THEN GO TO 6
   5 GO TO 7
   6 PRINT INK 3; FLASH 1;AT m,n;"K"
   7 GO TO 520
 210 IF INKEY$="o" THEN BEEP 0.001,45: GO TO 225
 220 GO TO 290
 225 IF d=0 THEN GO TO 3040
 230 IF ATTR (c+1,d-1)<>8 THEN GO TO 250
 240 GO SUB 2000: LET c$="I": LET b$="G": LET d=d-1: GO SUB 2010: GO TO 280
 260 IF ATTR (c+1,d-1)=14 THEN GO TO 240
 270 IF ATTR (c,d-1)=13 THEN BEEP 0.4,30: LET pus=pus-2: GO SUB 7510: GO TO 2
 280 IF ATTR (c+2,d+1)=14 OR ATTR (c-1,d+1)=14 THEN PRINT INK 6;AT c,d+1;"M";AT c+1,d+1;"M": GO TO 430
 310 IF INKEY$="p" THEN BEEP 0.001,45: GO TO 325
 320 GO TO 390
 325 IF d=31 THEN GO TO 3020
 330 IF ATTR (c+1,d+1)<>8 THEN GO TO 350
 340 GO SUB 2000: LET b$="A": LET c$="D": LET d=d+1: GO SUB 2010: GO TO 380
 350 IF ATTR (c+1,d+1)=14 THEN GO TO 340
 370 IF ATTR (c,d+1)=13 THEN BEEP 0.4,35: LET pus=pus-2: GO SUB 7510: GO TO 2
 380 IF ATTR (c+2,d-1)=14 OR ATTR (c-1,d-1)=14 THEN PRINT INK 6;AT c,d-1;"M";AT c+1,d-1;"M": GO TO 430
 400 IF INKEY$="u" OR INKEY$="j" THEN GO TO 500
 410 IF INKEY$="0" THEN GO SUB 600
 430 GO SUB pas
 433 FOR f=0 TO 2: IF ATTR (c+2,d)=8 THEN GO SUB 2000: BEEP 0.01,35: LET c=c+1: GO SUB 2010: NEXT f: BEEP 0.5,-23: LET pus=pus-2: GO SUB 7510: GO TO 2:
 440 GO TO 100
 500 IF INKEY$="u" THEN BEEP 0.01,25: GO TO 560
 510 IF INKEY$="j" THEN BEEP 0.01,30: GO TO 540
 520 GO SUB pas: IF ATTR (c+2,d+1)=12 OR ATTR (c+2,d-1)=12 THEN GO TO 420
 530 GO TO 500
 540 IF ATTR (c+2,d)=14 THEN LET c=c+1: GO SUB 2010: PRINT INK 6;AT c-1,d;"M"
 545 IF c=19 THEN GO TO 3010
 550 GO TO 520
 560 IF ATTR (c-1,d)=14 THEN LET c=c-1: GO SUB 2010: PRINT INK 6;AT c+2,d;"M"
 565 IF c=1 THEN GO TO 3000
 570 GO TO 520
 600 IF ATTR (c+2,d)=14 OR ATTR (c-1,d)=14 THEN RETURN
 605 IF b$="A" THEN GO TO 700
 606 IF d<=3 THEN RETURN
 610 RESTORE 635: FOR f=0 TO 7 STEP 2: BEEP 0.01,c+20: READ x: READ y
 625 IF ATTR (c+x+1,d+y)=12 THEN RETURN
 626 GO SUB pas: IF ATTR (c,d)<>15 OR ATTR (c+1,d)<>15 THEN GO TO 755
 630 GO SUB 2000: LET c=c+x: LET d=d+y: IF ATTR (c,d)<>8 OR ATTR (c+1,d)<>8 THEN GO TO 740
 631 GO SUB 2010
 632 IF ATTR (c-1,d)=139 THEN GO TO 800
 633 NEXT f: RETURN
 635 DATA -1,-1,-1,-1,1,-1,1,-1
 706 IF d>=28 THEN RETURN
 710 RESTORE 735: FOR f=0 TO 7 STEP 2: BEEP 0.01,c+20: READ x: READ y
 725 IF ATTR (c+x+1,d+y)=12 THEN RETURN
 726 GO SUB pas: IF ATTR (c,d)<>15 OR ATTR (c+1,d)<>15 THEN GO TO 755
 730 GO SUB 2000: LET c=c+x: LET d=d+y: IF ATTR (c,d)<>8 OR ATTR (c+1,d)<>8 THEN GO TO 740
 731 GO SUB 2010
 732 IF ATTR (c-1,d)=139 THEN GO TO 800
 734 NEXT f: RETURN
 735 DATA -1,1,-1,1,1,1,1,1
 740 IF ATTR (c,d)=12 OR ATTR (c+1,d)=12 OR ATTR (c+1,d)=14 THEN LET c=c-x: LET d=d-y: GO SUB 2010: GO TO 760
 750 GO SUB 2010: BEEP 0.4,34: GO SUB 7510: LET pus=pus-2: GO TO 2
 755 IF ATTR (c+x,d+y)=12 OR ATTR (c+1+x,d+y)=12 OR ATTR (c+1+x,d+y)=14 THEN GO TO 760
 756 GO SUB 7510: LET pus=pus-2: BEEP 0.4,34: GO TO 2
 760 FOR f=0 TO 1: BEEP 0.01,35: LET c=c+1: IF ATTR (c+1,d)<>8 THEN LET c=c-1: RETURN
 762 LET c=c-1: GO SUB 2000: LET c=c+1: GO SUB 2010: NEXT f: RETURN
 800 PRINT AT c-1,d;" ": LET i$((pis/10)+1)="a": LET du=du+1: GO SUB 7030: IF du=20 THEN GO TO 7400
 810 GO TO 760
2000 PRINT AT c,d;" ";AT c+1,d;" ": RETURN
2010 PRINT INK 7;AT c,d;b$;AT c+1,d;c$: RETURN
3000 LET pis=pis-50: LET c=18: GO TO 2
3010 LET pis=pis+50: LET c=1: GO TO 2
3020 LET pis=pis+10: LET d=0: GO TO 2
3040 LET pis=pis-10: LET d=30: GO TO 2
```

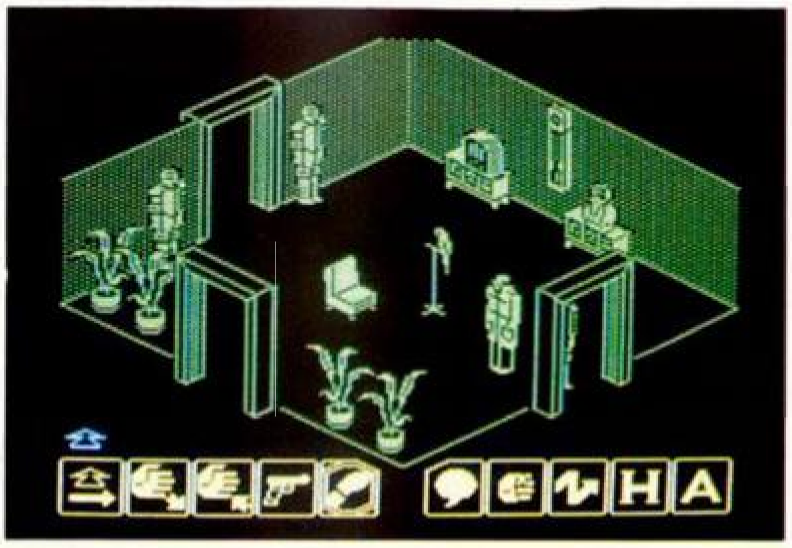


```
4000 LET m=4: LET n=15: RETURN
4010 LET m=13: LET n=5: RETURN
4020 LET m=4: LET n=25: RETURN
4030 LET m=6: LET n=9: RETURN
4040 LET m=8: LET n=22: RETURN
4050 LET m=9: LET n=14: RETURN
4060 LET m=11: LET n=11: RETURN
4070 LET m=12: LET n=20: RETURN
4080 LET m=7: LET n=21: RETURN
4090 LET m=9: LET n=9: RETURN
4100 LET m=8: LET n=17: RETURN
4110 LET m=5: LET n=25: RETURN
4120 LET m=5: LET n=14: RETURN
4130 LET m=12: LET n=15: RETURN
4140 LET m=10: LET n=28: RETURN
4150 LET m=12: LET n=20: RETURN
4160 LET m=13: LET n=16: RETURN
4170 LET m=5: LET n=2: RETURN
4180 LET m=13: LET n=22: RETURN
4190 LET m=4: LET n=11: RETURN
5010 PRINT INK 2;AT 1,3;"F";AT 1,10;"F";AT 1,17;"F";AT 1,24;"F";AT 2,3;"E";AT 2,10;"E";AT 2,17;"E";AT 2,24;"E": LET po=3: LET kl=0
5020 LET op=3: LET pas=pas+20
5060 LET po=po+1
5062 IF ATTR (c-1,d)=10 THEN LET pus=pus-2: BEEP 0.3,34: GO SUB 7510: GO TO 2
5064 IF ATTR (po,op)<>8 AND ATTR (po,op)<>10 THEN LET op=op+7: LET po=3
5070 IF kl=1 THEN PRINT AT po,op;" ": IF op>24 THEN LET op=3: LET kl=0: RETURN
5080 IF kl=0 THEN PRINT INK 2;AT po,op;"C": IF op>24 THEN LET op=3: LET kl=1: RETURN
5090 RETURN
5100 LET kl=0: LET po=1: LET op=1: LET pas=pas+20: GO TO 5120
5110 LET kl=0: LET po=16: LET op=25: LET pas=pas+10
5120 IF po=c AND op=d THEN BEEP 0.5,34: LET pus=pus-2: GO SUB 7510: GO TO 2
5121 PRINT AT po,op;" ";AT po+1,op;" "
5130 IF kl=0 THEN LET op=op-1: IF op=0 OR ATTR (po,op)=14 THEN LET kl=1
5140 IF kl=1 THEN LET op=op+1: IF op=30 OR ATTR (po,op)=14 THEN LET kl=0
5150 PRINT INK 5;AT po,op;"N";AT po+1,op;"O"
5160 IF po=c AND op=d THEN BEEP 0.5,34: LET pus=pus-2: GO SUB 7510: GO TO 2
5170 RETURN
5200 LET po=5: LET op=1: LET pas=pas+20: GO TO 5220
5210 LET po=17: LET op=25: LET pas=pas+10
5220 IF po=c+1 AND op=d THEN BEEP 0.5,34: LET pus=pus-2: GO SUB 7510: GO TO 2
5222 PRINT AT po,op;" "
5230 LET op=op-1: IF op=0 THEN LET op=29
5235 IF ATTR (po,op)<>8 AND ATTR (po,op)<>15 THEN LET op=29
5250 PRINT INK 3;AT po,op;"L"
5260 IF po=c+1 AND op=d THEN BEEP 0.50,34: LET pus=pus-2: GO SUB 7510: GO TO 2
5270 RETURN
5310 LET kl=0: LET po=8: LET op=6: LET pas=pas+10
5320 IF kl=0 THEN GO TO 5350
5330 IF kl=1 THEN GO TO 5340
5340 PRINT AT po,op+3;" ";AT po+9,op+3;" ": LET op=op-1: IF op=4 THEN LET kl=0: RETURN
5345 GO TO 5380
5350 PRINT AT po,op;" ";AT po+9,op;" ": LET op=op+1: IF op=23 THEN LET kl=1: RETURN
5355 GO TO 5380
5380 PRINT INK 4;AT po,op;"BHBH";AT po+9,op;"BHBH": LET pas=5390: RETURN
5390 LET pas=5320: RETURN
5400 LET po=5: LET op=4: LET pas=pas+20: GO TO 5420
5410 LET po=5: LET op=5: LET pas=pas+10
5420 PRINT AT po,op;"    ": LET po=po+1: IF po=20 THEN LET po=3
5430 PRINT INK 4;AT po,op;"BHBH": LET pas=5440: RETURN
5440 LET pas=5420: RETURN
5510 LET po=3: LET op=4: LET pas=pas+10
5520 PRINT AT po,op;" ";AT po,op+5;" ";AT po,op+10;" ": LET po=po+1: IF po=19 THEN LET po=4
5530 PRINT INK 5;AT po,op;"E";AT po,op+5;"E";AT po,op+10;"E": RETURN
7000 IF pus=0 THEN GO TO 7300
7020 PRINT #0;AT 1,0;"VIDAS";AT 0,7;n$(1 TO pus);AT 1,7;m$(1 TO pus)
7030 PRINT #0;AT 1,15;"OBJETOS ";du: RETURN
7310 PRINT INK 4;AT 19,0;r$;s$;t$;AT 17,15;"B";AT 18,15;"C"
7320 FOR f=0 TO 16: BEEP 0.1,f+20: PRINT INK 4;AT f,0;r$;s$;t$: NEXT f
7330 PRINT PAPER 4;AT 3,7;"SE ACABARON TUS DIAS";AT 10,1;"PULSA - Q - TECLA PARA EMPEZAR"
7340 IF INKEY$="q" THEN GO TO 1
7350 IF INKEY$<>"" THEN STOP
7360 GO TO 7340
7410 CLS : RESTORE 9910: GO SUB 9000
7420 GO SUB 9990: PRINT INK 6;AT 14,15;"ABC";AT 15,15;"DEF";AT 16,15;"GHI";AT 17,15;"JKL"
7430 NEXT f
7440 PRINT #0;AT 0,10;"LO CONSEGUISTE";AT 1,9;"-Q- OTRA PARTIDA": GO TO 7340
7500 LET ox=c: LET oy=d: RETURN
7510 LET c=ox: LET d=oy: RETURN
8000 LET o$="          MARK es un intrepido aventurero,ayudale a conseguir el tesoro del gran templo,para ello tienes que coger las 20 llaves    Arriba...U  Abajo...J  Derecha...P  Izquierda...O  Salto...0"
8010 RESTORE 8050: FOR f=0 TO 5: READ x
8020 FOR y=0 TO 8
8030 PRINT INK 4;AT 3+y,x;"██"
8040 NEXT y: NEXT f
8050 DATA 1,7,10,14,17,25
8060 LET v$="██": LET w$="": LET x$="": RESTORE 9900: GO SUB 9000: PAUSE 200
8070 FOR f=1 TO 189: LET p=INT (RND*7): BORDER p: BEEP 0.02,p: PRINT INK p;AT 16,0;o$(f TO f+31): NEXT f: BORDER 0: RETURN
9010 LET a=3: LET b=0: LET e=9300: LET h=9400: READ g: READ pas: READ a$
9020 FOR f=1 TO g: LET p=CODE a$(f): IF b=1 THEN GO TO 9200
9050 IF b=2 THEN GO TO e
9060 IF b=3 THEN GO TO h
9080 IF a$(f)="0" THEN LET a=a+1: NEXT f
9090 IF a$(f)="#" THEN LET b=2: NEXT f
9100 IF a$(f)="$" THEN LET b=3: NEXT f
9120 FOR l=1 TO 9: IF CODE a$(f)=l+48 THEN LET v=l
9140 NEXT l
9150 LET b=1: NEXT f: RETURN
9200 IF a$(f)="1" THEN LET v$="███": LET w$="B█D": LET x$=" E": GO TO 9260
9210 IF a$(f)="2" THEN LET v$="█████": LET w$="B█HJ": LET x$=" E": GO TO 9260
9220 IF a$(f)="3" THEN LET v$="██████": LET w$="B█HJE": LET x$="E": GO TO 9260
9230 IF a$(f)="4" THEN LET v$=r$: LET w$=s$: LET x$=t$: GO TO 9260
9250 LET p=p-95: PRINT INK 4;AT a,p;v$;AT a+1,p;w$;AT a+2,p;x$
9260 LET v=v-1: IF v=0 THEN LET b=0: LET a=a+1
9270 NEXT f
9300 LET u=p: LET e=9310: NEXT f
9310 LET v=p: LET e=9320: NEXT f
9320 LET w=p: LET e=9300: LET v=v-95: LET u=u-95: LET w=w-95: LET b=0: FOR n=0 TO u: PRINT INK 6;AT v+n,w;"M": NEXT n: NEXT f: RETURN
9400 LET x=p: LET h=9410: NEXT f
9410 LET y=p: LET h=9400: LET x=x-95: LET y=y-95: LET b=0
9420 GO SUB 9990: IF x<>0 THEN GO TO 9440
9430 PRINT INK 5;AT y-2,x;"M";AT y-1,x;"N";AT y,x;"O": GO SUB 9980: NEXT f: RETURN
9440 PRINT INK 5;AT y-2,x;"P";AT y-1,x;"Q";AT y,x;"R": GO SUB 9980: NEXT f: RETURN
9700 DATA 43,5210,"00021_22b21!53fkpu00000006fejouz#col$_p$_d"
9710 DATA 36,5510,"0000062_di1n01q01l32w{00009_dix3ns1!"
9720 DATA 38,5010,"00000043puz1j022e23_000091m{!2_fsx#ify"
9730 DATA 37,5110,"00063kpzu1001_06b2p3fk02tz000024_#ecp"
9740 DATA 41,5010,"00024_00000053_zu400001_#l{x#{o}$"d$"k$"p"
9750 DATA 39,5200,"00024_000000051h3kp002uz#gkh#gkt$_d#d_l"
9760 DATA 36,5010,"00063_2g1d2kn43puz00000041clu5_hgy!"
9770 DATA 30,5310,"0000042_3z000000003z2_#iea#dn!"
9780 DATA 31,5410,"0000023_01h052ru3m00000hmrw1!2_"
9790 DATA 39,9980,"00000021!1x1t1p1l1h1{01_#hjk#g_}#lf{$"g"
9800 DATA 24,9980,"0000033tz00002hm#k_h#f_l"
9810 DATA 33,5510,"0000052_din05{1f3v00002uy#egu#fl}"
9820 DATA 28,9980,"000000024_00000032w{#h_!#dnv"
9830 DATA 26,5010,"000000024_00000091_w2chmf{"
9840 DATA 39,5400,"000000021_002gj02nw2r{1_#k_k#e_{$"n#em}"
9850 DATA 33,5310,"000043z2_00000000003_3z#kda$_e$_p"
9860 DATA 37,5110,"000022_1d2is51!2nx00000007_dinsw{#h_}"
9870 DATA 35,9980,"000041vy!001_0022s1n3x1!22i1d1_#e_v"
9880 DATA 32,5400,"000062hns1_1x1!000001_007h2kpu3z"
9890 DATA 29,5210,"00000042_cg0000000024_#p_}$"p"
9900 DATA 32,1,"5aekr!3ct!2t{3ktz2ry2sz2t{2t!2t!"
9910 DATA 31,1,"32jp02ft02bx002bx002bx02ft002ko"
9920 RESTORE 9930: FOR a=USR "a" TO USR "s"+7: READ b: POKE a,b: NEXT a: RETURN
9930 DATA 0,7,63,60,60,16,16,124,255,127,127,63,7,7,7,3,32,48,24,12,12,24,48,32,124,228,228,244,176,144,144,216,255,126,124,124,124,56,16,0,255,189,153,195,231,255,255,255,0,224,252,60,60,8,8,60,255,254,252,248,192,192,192,128,62,39,39,47,13,9,9,27
9940 DATA 255,62,28,12,8,0,0,0,80,136,80,32,20,8,5,2,0,32,97,255,225,97,32,0,129,129,129,129,129,129,129,255,8,28,62,107,107,127,54,20,1,3,6,13,27,55,111,223,0,0,0,0,8,28,118,231,28,73,107,85,8,85,99,85,219,189,126,255,255,255,255,255,128,192,96,176,216,236,246,251
9950 RESTORE 9960: POKE 23675,176: POKE 23676,254: FOR a=USR "a" TO USR "r"+7: READ b: POKE a,b: NEXT a: RETURN
9960 DATA 0,1,3,7,15,31,28,62,126,255,255,255,255,255,126,60,0,128,192,224,240,248,120,252,63,115,237,232,233,243,252,255,153,195,255,126,189,126,255,255,252,208,183,23,151,207,63,255,255,255,255,255,254,124,124,63,255,255,255,73,73,73,73,255,255,255,255,255,127,62,62,248,24,28,12,7,7,3,1,0,73,73,73,73,255,255,255,126,24,56,48,224,224,192,128,0
9970 DATA 248,248,252,206,206,252,248,248,240,240,248,254,238,226,224,240,248,248,192,192,200,248,240,224,31,31,63,115,115,63,31,31,15,15,31,127,119,71,7,15,31,31,3,3,19,31,15,7
9980 POKE 23675,88: POKE 23676,255: RETURN
9990 POKE 23675,176: POKE 23676,254: RETURN
```

MOVIE. • Videoaventura • Imagine

EL DETECTIVE MARLOW

Todos los días no tenemos la oportunidad de disfrutar en nuestro ordenador de un programa como éste. Movie es un juego que, tanto por sus gráficos como por su gran originalidad, está llamado a ocupar el lugar de una auténtica estrella.



En esto de los juegos para ordenador, las sorpresas son inagotables. Cuando aparece un superprograma, como bien pudieran ser los casos de Alien 8 o Fairlight, parece que va a ser casi imposible que surjan otros nuevos que los superen o incluso, igualen en su elevado nivel de calidad.

Sin embargo, inesperadamente y sin saber cómo ni por qué, cada pocos meses aparece en el mercado un



juego que vuelve a despertar la admiración de todos y que deja boquiabiertos a los usuarios.

Movie es uno de estos casos. Movie, con su excelente técnica de programación y con sus magníficas innovaciones, supone un paso adelante en este ya largo camino del software, y posiblemente, tras su inminente lanzamiento en nuestro país, no tardará mucho tiempo en alcanzar los primeros puestos en las listas de éxitos.

Pero vayamos por partes y contemos, antes de nada, en qué consiste el argumento del programa.

Años 30. Nueva York. Jack Marlow, brillante y afamado detective privado, afronta uno de los casos más importantes de los últimos tiempos, relacionado con el siempre desagradable y peligroso asunto del Hampa. La misión: adentrarse en la guarida de Bugs Maloy, el rey de la mafia, encontrar entre sus múltiples estancias una cinta de cassette que contiene una valiosísima información y regresar a escucharla en su oficina.

Pero una guarida de gánsters no es un parque de atracciones, y los peligros aguardan en cada esquina, tras cada puerta, en cada habitación. Por tanto, Marlow tendrá que permanecer siempre alerta, y además de ir recorriendo todos los lugares que le sea posible para encontrar los objetos que le sirvan de ayuda a la hora de completar su misión, tendrá que vigilar en todo momento los movimientos de los numerosos guardianes que custodian a Bugs Maloy, quiénes dispararán o se avalanzarán sobre él al menor descuido.

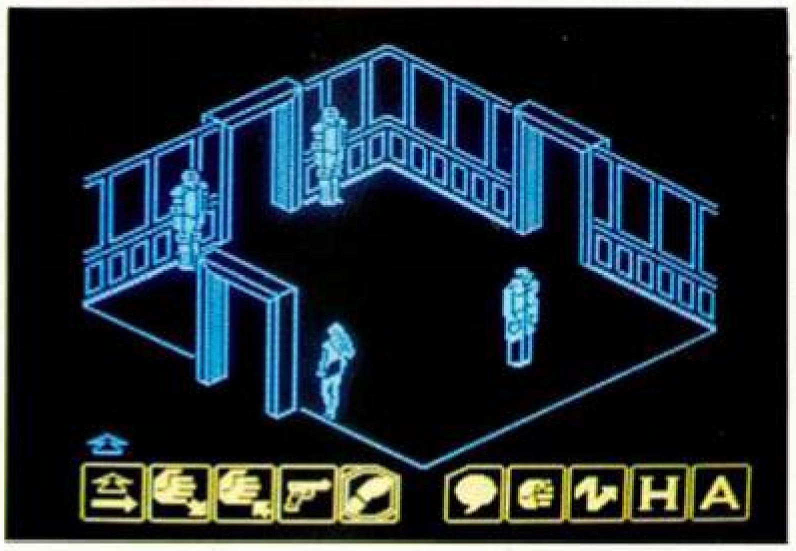
Sin embargo, afortunadamente no todo son enemigos. También existe una buena cantidad de confidentes y amigos que están dispuestos a servirnos de ayuda. Y aquí radica uno de los puntos más importantes de este programa: en el hecho de que podemos dialogar libremente con estos personajes con el fin de sonsacarles la información necesaria para poder continuar con la aventura. Estos nos facilitarán las claves que nos permitirán abrir puertas en principio inaccesibles, o algunas otras pistas de utilidad. Pero siempre que conversemos con ellos deberemos ser cautos y tratarlos correctamente, o de lo contrario nos ignorarán por completo.

Pero con todo esto, es decir, con el uso correcto de determinados objetos y con el conocimiento de ciertas claves, aún no podremos llegar hasta el despacho de Bugs, pues la única persona que sabe su paradero es la joven y bella Tanya. Cuando Marlow consiga dar con ella, podremos decir que sus problemas habrán acabado. Pero ¡cuidado!, Tanya tiene una hermana gemela llamada Vanya, pero que trae unas intenciones bien distintas. A simple vista nos resultará imposible diferenciarlas, pero Vanya intentará confundirnos a toda costa y si seguimos sus pasos nos conducirá a una muerte segura.

Esto es, en líneas generales, el argumento de Movie, aunque, lógicamente, a la hora de jugar con él, da mucho más de sí y posiblemente tardaremos bastantes semanas en completarlo.

Y si el tema es ya más que interesante de por sí, los aspectos gráficos del mismo lo son tanto o más. De momento decir que posee más de doscientas cincuenta pantallas (casi nada), y que todas ellas presentan una calidad inmejorable, pues además de estar realizadas con todo lujo de detalles, desde el punto de vista meramente artístico, resultan de lo más atractivo visto hasta la fecha.

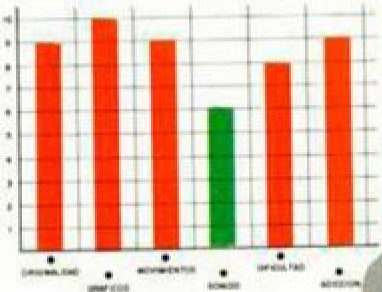
A pesar de que todas ellas están realizadas a un solo color, no desmerece en absoluto del resultado final, y la verdad es que si encima pudiéramos utilizar más

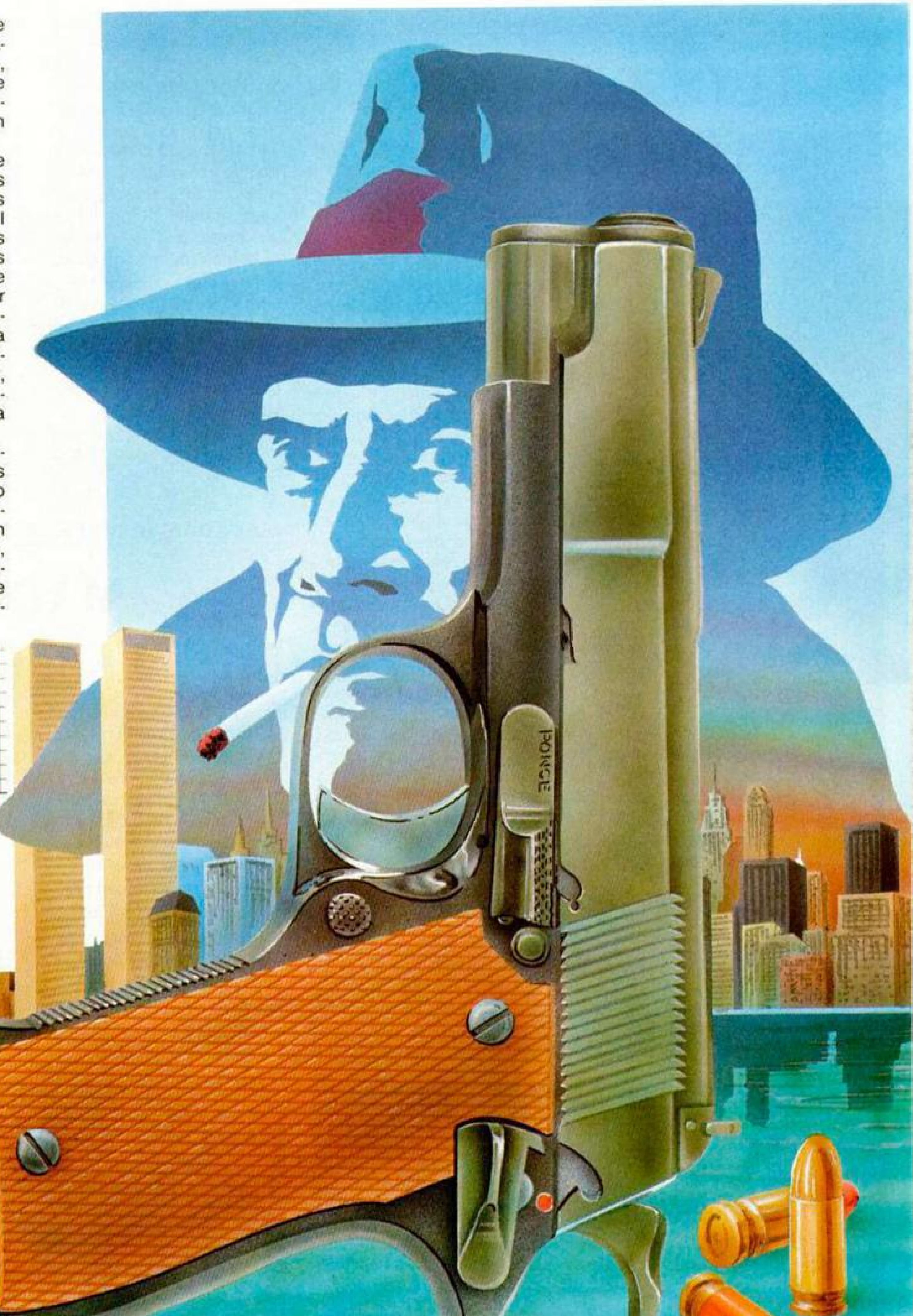


colores, muy posiblemente necesitaríamos un ordenador nuevo y más potente, pues ya con los que tiene este programa, parece imposible que quepa en tan sólo 48K de memoria.

Otro detalle destacable de este Movie es que los movimientos y acciones que puede llevar a cabo el personaje, los controlamos muy cómodamente gracias a un sistema de iconos que aparece en la parte inferior de la pantalla. Con él podemos acceder a cualquiera de las diez opciones diferentes: inventario, soltar, coger, disparar, andar, hablar, golpear, arrojar, pausa y anular el juego.

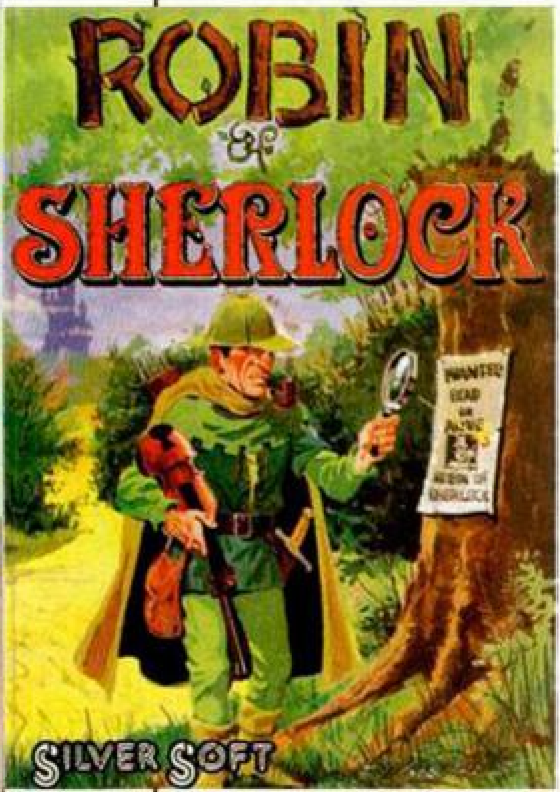
Como podréis comprobar, realizar un juego más completo debe ser poco menos que imposible, y encontrar un programa con un argumento tan interesante, unos gráficos tan magistrales y con tanta variedad de acciones, va a resultar verdaderamente difícil.





ROBIN OF SHERLOCK. Aventura. Silver Soft.

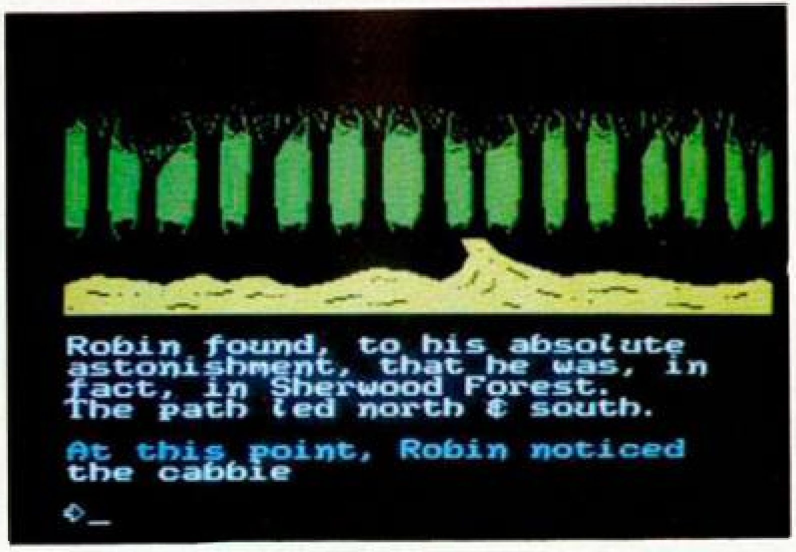
EL CASO DE LOS TAXIS Y LOS VIOLINES



No cabe duda de que las aventuras de texto son un tipo de programas que no despiertan demasiado interés en nuestro país. Sin embargo, llama la atención el hecho de que en Gran Bretaña —hoy por hoy el santuario del software—, estos juegos son considerados como algo muy especial y cuentan con un elevado número de adictos.

¿A qué es debida esta diferencia tan abismal? Evidentemente a los motivos derivados de un idioma distinto, lo cual llega a ser insalvable en muchos casos, pero no al hecho de que estos programas resulten malos o aburridos, ya que se ha podido comprobar que aventuras como Gremlins o Misión 1, que fueron traducidas al castellano, alcanzaron un aceptable éxito en nuestro país.

Por esta razón, tendríamos que darles un pequeño tirón de orejas a las casas que se encargan de la importación de juegos, las cuales no se toman demasiadas molestias a la hora de traer sus productos y simplemente se limitan a traducir las instrucciones en lugar de realizar una verdadera versión española de dichos programas. Por esta razón hoy en día, afortunada o desgraciadamente, nadie desconoce el significado de frases como: 'Pres fire to start', 'Redifine keys' o 'You solved 100 por 100 of game'.

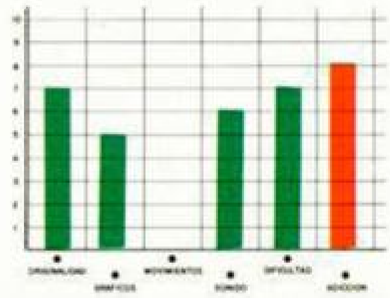


Así pues, afirmar que Robin of Sherlock es una aventura original, realizada con un gran sentido del humor y con la cual la diversión está prácticamente asegurada, es como predicar en el desierto, puesto que sabemos perfectamente que este juego es muy difícil que llegue hasta nuestro país, y mucho menos traducido.

Sin embargo, la misión principal de esta sección de «Nuevo», es la de hacernos eco de todas las novedades que se producen en el mercado y este juego forma parte, como el que más, de la actualidad del software.

En esta aventura representamos el papel de un Sherlock Holmes muy especial que intentará desentrañar un interesante caso relacionado con robos, taxis, violines, y muchas otras cosas más que complican bastante la situación y que iremos descubriendo a medida que vayamos controlandola.

Como antes dijimos, Robin of Sherlock es un programa muy interesante y divertido que, salvo por el detalle de que sus gráficos no son demasiado brillantes, ocupa un lugar destacado dentro de la amplia lista de aventuras de texto.



MACADAM BUMPER. Simulador. PSS-ERE.

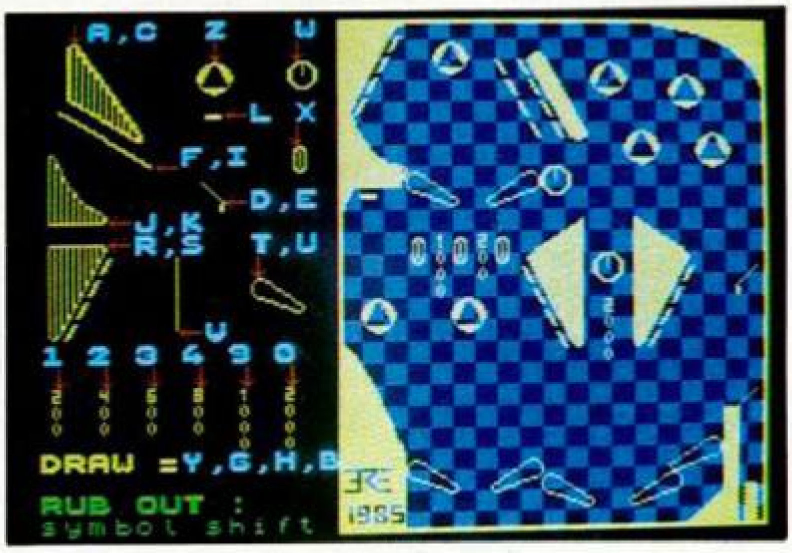
UNA «MAQUINITA» MUY ESPECIAL

Los Pin-ball se han vuelto a poner de moda en los últimos meses. La verdad es que no es de extrañar debido a que este tipo de máquinas resultan de lo más adictivo y entretenido.

Ahora, con este Macadam-Bumper, existe la posibilidad de tener uno de estos pin-ball en nuestra casa, y podremos jugar todas las partidas que nos apetezca sin echar cada dos por tres cinco duros a la maquinita.

El ponernos ahora a explicaros en qué consiste este tipo de juegos, sería un poco absurdo ya que cualquiera conoce más que de sobra la 'ciencia' de ellos.

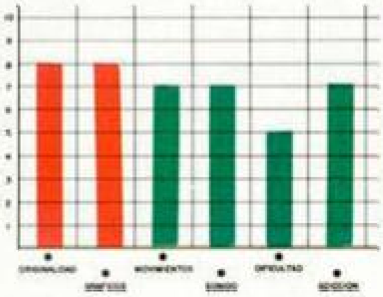
Por otra parte, también es muy posible que os estéis preguntando que a cuento de qué os venimos ahora con un juego como éste del cual existen actualmente cantidad de variantes en el mercado y que prácticamente no representan ninguna novedad. Sin



embargo, Macadam, en contra de lo que puediera parecer, sí que es un programa innovador en el sentido de que te puedes crear tu propia máquina.

Este es sin duda su mayor atractivo, ya que además de poder jugar con el modelo que ya viene en el propio programa, puedes acceder a un completo menú de opciones que te permitirán elegir entre una considerable cantidad de elementos para componer a tu antojo la máquina que más te guste y a la que podrás modificar a tu manera siempre que lo desees, demostrando así, tu capacidad de creación.

Por tanto, Macadam no es sólo un juego de pin-ball, sino que es un juego de millones de pin-ball.





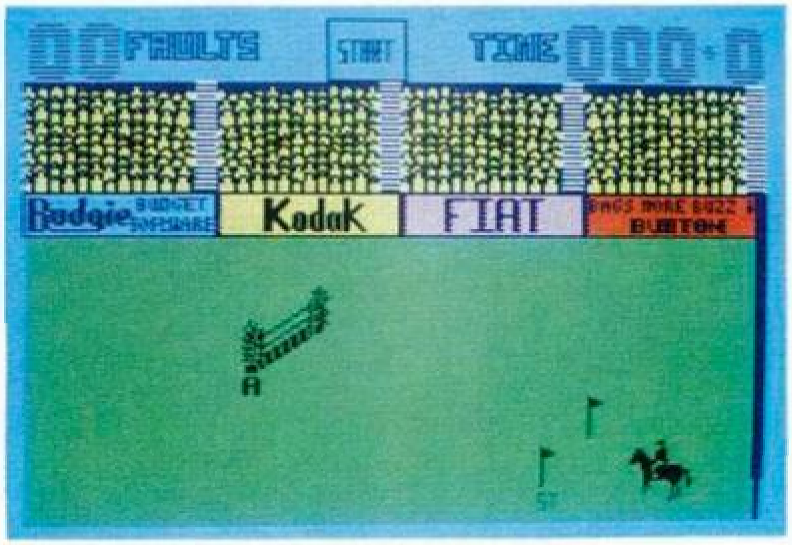
SHOW JUMPING. Deportivo. Alligata.

UNA TARDE EN LA HIPICA

Un nuevo programa de simulación deportiva, esta vez de hípica, llega hasta nuestros ordenadores a través de Alligata Soft.

El juego transcurre en un verde y concurrido hipódromo y representa el desarrollo de una competición de saltos de obstáculos a caballo. En él, el participante o participantes, (hasta un máximo de ocho), deben ir completando correctamente una serie de recorridos compuestos por diferentes cantidades de obstáculos intentando realizar el menor número posible de derribos. Además, tal y como ocurriría en una competición real, en la puntuación final influirá también el hecho de conseguir finalizar la prueba dentro del período de tiempo establecido para cada una de ellas.

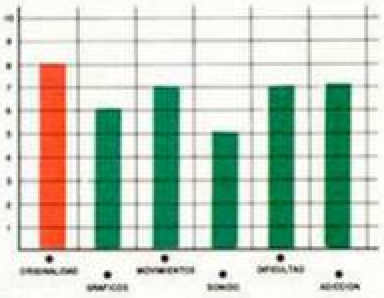
Los circuitos vienen indicados, además de por un pequeño esquema que aparece en pantalla inmediatamente antes de comenzar la competición, por una letra que aparece junto a cada obstáculo, las cuales, si las seguimos en orden alfabético, nos irán indicando qué camino debemos recorrer.

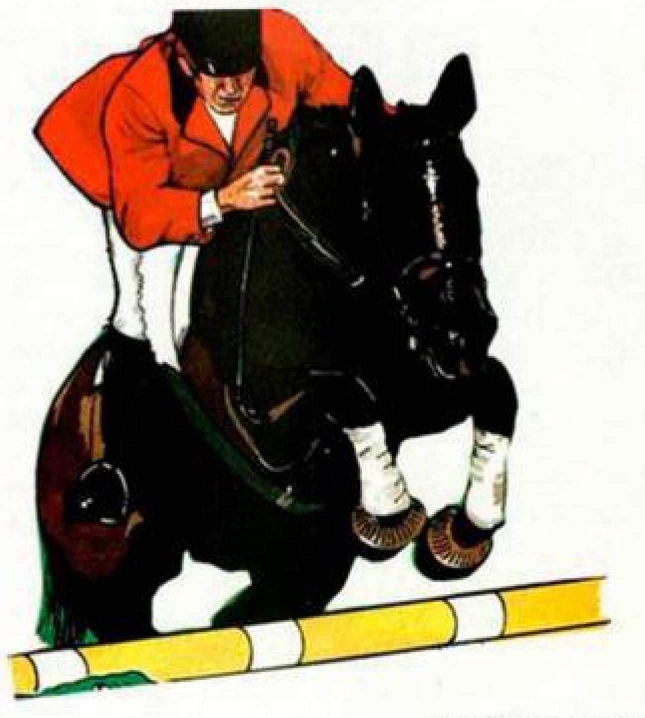


El tiempo comienza a correr cuando atravesamos la línea de salida y acaba justo en el momento en el que hacemos lo propio entre los banderines de salida.

Este Show Jumping es un juego que al principio resulta bastante complicado, ya que el hacerse con el manejo y buen control del caballo no es excesivamente sencillo, pues éste puede girar en múltiples direcciones y los obstáculos deben ser afrontados desde el ángulo adecuado. Si no lo hacemos así, el caballo rehusará a saltar, por lo que tendremos que volver sobre nuestros pasos e intentar situarnos correctamente llevando la velocidad necesaria. Si hacemos que el caballo rehúse tres veces durante el desarrollo de una prueba, seremos automáticamente descalificados.

Como veréis, el programa es bastante real en cuanto a las reglas que lo rigen y en lo referente a los movimientos del caballo, por lo que, si exceptuamos la apariencia gráfica del mismo, que resulta un tanto pobre, en conjunto Show Jumping es un juego de una calidad aceptable y que resulta bastante divertido una vez que aprendemos a ser experimentados jinetes.





"EL SACRIFICIO DE AVARATH Y EL TRIUNFO DEL UNICORNIO"

Diego GOMEZ

Ha llegado la hora que todos temíamos: AVARATH va a entregar su vida para que las fuerzas del bien, una vez más, vuelvan a triunfar.

Toma a AVARATH y ve a Harvena. Recoge el libro y con ello volverás a KIELMATH a la normalidad. Entrégale el libro y su bastón y haz que le devuelva a AVARATH el suyo. Deja que el mago recupere fuerzas con la ayuda de ULIN-GAIL y dirígete luego hacia Hail-An-Hes donde, con el cayado, podrás romper la piedra y descubrir el pergamino de la salud (Scroll of Healing). Déjalo allí y encamínate a Galzar de donde tomarás «The Horn of Summons». Ve de nuevo hasta Amorlet y deja allí a AVARATH.

Dirígete a Oronoman y coge la jarra blanca de entre las tres que allí se encuentran. Con ella en tu poder ve al nacimiento del río Derris y llena la jarra con el espíritu del agua. Lleva este espíritu hasta Amorlet y déjalo allí.

Elige ahora cualquiera de las dos jarras en Oronoman y ve a Clarooth de donde tomarás el espíritu del viento. Deja este espíritu también en Amorlet.

Con SHARMEK, ROLQUIN o LAIRMATH toma la última de las jarras y ve a la ciudad de Rimersel donde se encuentra el espíritu de la tierra. Llévalo hasta Amorlet y déjalo allí.



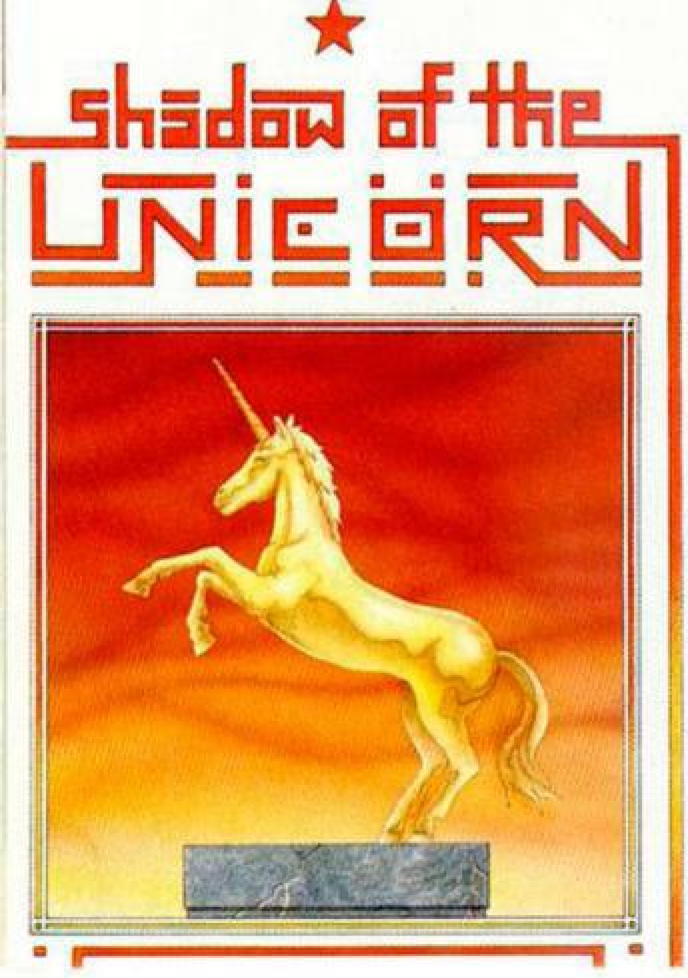
Raquín es uno de los tres personajes que pueden ir a Rimersel en busca del espíritu de la tierra.



Aquí se encuentra la piedra que Avarath romperá con su cayado para descubrir el pergamino de la salud.



Holdin se encamina hacia Marith en busca de la lámpara y el pergamino de la salud.



KIELMATH va hasta Monar y coge allí la Esfera del Día (The orb of the Day). Con ella y con el libro te diriges a la pirámide de Amarnil.

Toma a VILYAN y abandona el bosque de la muerte. Ve hasta Brantiem (El corazón oscuro) y en la profundidad de este bosque encontrarás la Pluma de KIELMATH (Kielmath's Kill). Tómala y llévala a Amarnil.

Con GUINOL o HOLDIN ve a Marith donde hay una lámpara. Tómala y recoge en Hail-An-Hes el Pergamino de la Salud. Con ambas cosas dirígete a Hasan donde descubrirás la madriguera de Topil. Allí se encuentra la Esfera de la Noche (The Orb of the Night). Llévala a Amarnil.

Toma a AVARATH que se encuentra en Amorlet con los tres espíritus y asegúrate que tiene su cayado mágico y el cuerno de druida. Sitúale sobre el Agujero de Fuego en donde habrá de consumarse el sacrificio. (Ojo, el programa no está hecho por sádicos, por lo que el sacrificio sólo está representado por la musiquilla indicadora de que has hecho algo bien).

En Amarnil, KIELMATH, con la pluma y la Esfera de la Noche vuelve a sellar el libro encerrando dentro de él a todas las fuerzas maléficas.

En su pedestal de la isla de Oslar, el corazón de ILIS CLAIR, el UNICORNIO, late con fuerza. El guarda ahora la única llave capaz de abrir de nuevo el libro maldito y no hay fuerza física o mágica capaz de arrebatársela. Su sombra ampara de nuevo los reinos de FALFORN y ORONFAL.


SI BUSCAS LO MEJOR ERBE Software LO TIENE
MOVIE

MOVIE



Vamos a colocarlos en un determinado orden, y tendremos la dirección del octeto buscando:

```
BITE MAS SIGNIFICATIVO
Ø 1 Ø 14 13 s2 s1 sØ
FIJO  LINEA   SCAN
```

```
BYTE MENOS SIGNIFICATIVO
12 11 1Ø c4 c3 c2 c1 cØ
LINEA      COLUMNA
```

Es decir: de izquierda a derecha, los tres primeros bits son «Ø10» y este valor es fijo para todo el archivo. Los dos bits siguientes, son «l4» y «l3» es decir, los dos más altos de la línea. Los tres que siguen, son los que definen el scan, es decir, «s2», «s1» y «sØ». Aquí acaba el octeto más significativo. En el menos significativo, los tres primeros son los tres más bajos del número de línea «l2», «l1» y «lØ». Los cinco restantes indican el número de columna «c4» a «cØ». Ver Figura 9-16.

A la vista de esta ordenación, podemos sacar algunas conclusiones interesantes. Supongamos que tenemos en «HL» la dirección de un determinado octeto de pantalla que pertenecerá a un determinado carácter. Supongamos, también, que este octeto es el que corresponde al primer scan de la línea. Cada vez que incrementemos «H», pasaremos al siguiente scan del mismo carácter; por tanto, podremos incrementar «H» siete veces para cubrir los 8 scans de cada carácter. Esto será muy útil cuando desarrollemos una rutina para imprimir caracteres en pantalla.

Por otro lado, si incrementamos «L», pasaremos al siguiente carácter de la siguiente columna, pero, dentro de la misma línea, con lo que podremos imprimir una línea de un golpe, si direccionamos su primer carácter y vamos incrementando «L» 31 veces para cubir los 32 caracteres de la línea. Pero, ¿qué pasaría si incrementáramos «L» más allá de este punto? Para contestar esta pregunta, vamos a ver qué aspecto tienen los números de línea si los vemos en binario. Las líneas comprendidas entre Ø y 7 se verían: ØØØØØ a ØØ111. Las que estuvieran entre 8 y 15, se verían: Ø1ØØØ a Ø1111. Finalmente, las comprendidas entre 16 y 23 se verían: 1ØØØØ a 1Ø111.

Tal vez ya se haya dado cuenta de que los dos primeros bits «l4» y «l3» son «ØØ» para la primera zona de pantalla, «Ø1» para la segunda y «1Ø» para la tercera. De forma que podemos decir que estos bits nos están indicando la zona de pantalla a la que pertenece nuestro octeto. Por otro lado, los tres bits restantes («l2» a «lØ») nos indican el número de línea dentro de una determinada zona.

Si trabajamos dentro de una zona y sin salirnos de ella, podemos ir incrementando «L» y saltaremos de una línea a la siguiente sin ningún problema. Para cambiar de zona, no tendremos más que sumar 8 al registro «H» y nos encontraremos en la misma línea de la siguiente zona.

Hechas estas consideraciones, está claro que en la dirección de un determinado octeto están todos los datos que nos van a permitir saber a qué línea pertenece, a qué columna e, incluso, a qué scan del carácter, es decir, tenemos todos los datos que necesitamos sobre el octeto en cuestión. Sólo nos falta desarrollar una rutina que haga todo esto de forma automática.

De la misma forma que hicimos para el fichero de atributos, vamos a desarrollar una rutina que nos dé la dirección del primer scan de un carácter a partir de sus coordenadas. Hemos llamado «DIR» a la rutina. Se entra en ella con «DE» conteniendo las coordenadas del carácter en cuestión. Como siempre, «D» contiene el número de línea y «E» el número de columna. A la salida de la rutina, tendremos en «DE» la dirección del octeto perteneciente al primer scan de ese carácter. El listado, en Assembler, de «DIR» es el siguiente:

```
100 DIR   LD   A,D
110       AND  #7
120       RRC  A
130       RRC  A
140       RRC  A
150       OR   E
160       LD   E,A
170       LD   A,D
180       AND  #18
190       OR   #40
200       LD   D,A
210       RET
```

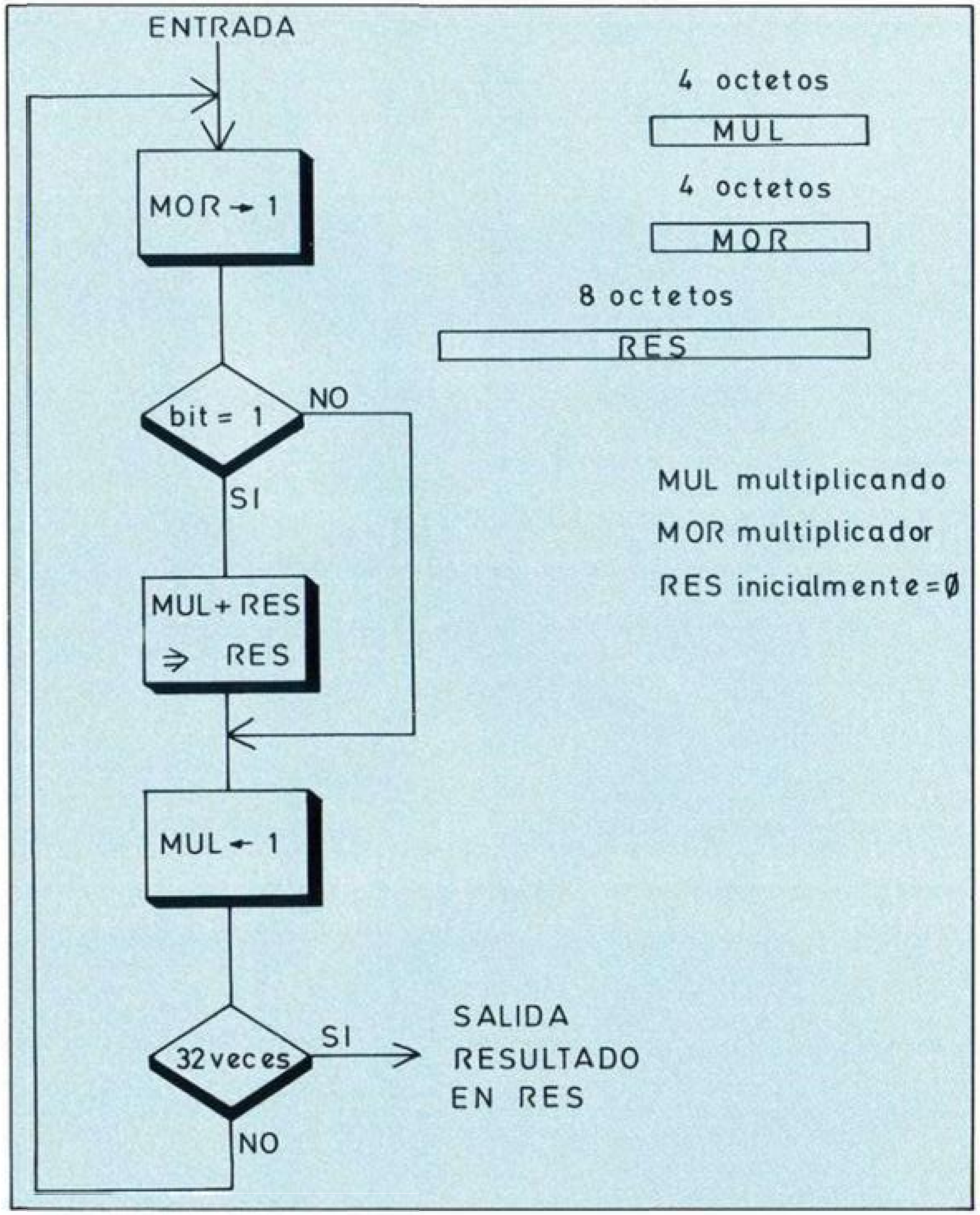



Fig. 9-14. Multiplicación por desplazamientos.

que ese número decimal está en un campo justificado a la derecha, para añadirle el cero es necesario desplazarle una posición hacia la izquierda:

```
      3 Ø
|-|-|-|-|-|
     3 Ø Ø
|-|-|-|-|-|
```

Por el mismo motivo si desplazamos un número binario una posición a la izquierda, o sea le añadimos un cero a la derecha, este número queda multiplicado por la base.

Cada uno de sus bits pasarían de valer 2 elevado a «e», a valer 2 elevado a («e» + 1):

```
00010        -2
!-!-!-!-!-!

00100        -4
!-!-!-!-!-!
```

Desplazarlo otra posición equivaldría a multiplicarlo otra vez por dos, esto es multiplicar el número original por 4 = 2 x 2; un nuevo desplazamiento sería 8 = 2 x 2 x 2 y así sucesivamente. Lo que se observa es que un nuevo desplazamiento implica multiplicar por la siguiente potencia de 2, esto es 2, 4, 8, 16, 32, etc. Por lo tanto, podríamos enunciar la siguiente regla: para multiplicar un número binario por una potencia de 2 se desplaza el número a la izquierda tantas veces como valor tenga el exponente.

```
NUMERO-BINARIO×2ᵉ=
=NUMERO-BINARIO→«e» veces
```

EJEMPLOS:

```
0001100 *2³=
=1100000;12×8=96

0011010*2²=
=1101000;26×4=104
```

Pero no siempre se quiere multiplicar un número por una potencia de dos, es más, la mayoría de las veces no es así.

En un micro-procesador que carece de instrucción de multiplicar, una forma inmediata de solucionar el problema es sumar el multiplicando tantas veces como valor tenga el multiplicador; por ejemplo 30 x 3 sería

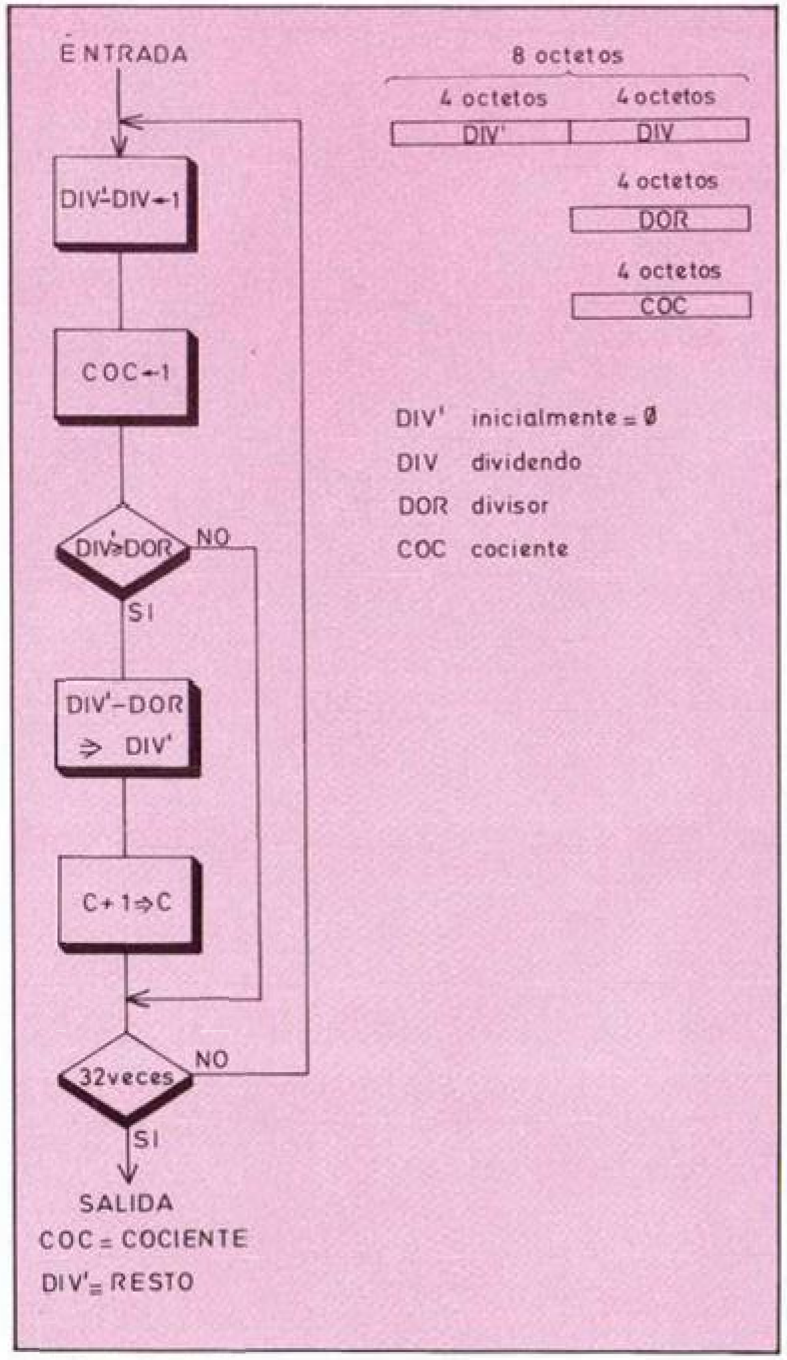


Fig. 9-15. División por desplazamientos.

30 + 30 + 30. En el caso de multiplicadores de poco valor, ésta sería la solución más sencilla, se ejecutarían poco más de tres instrucciones. Pero vamos a ver otra técnica que resulta mejor en el caso de multiplicadores grandes.

Como se sabe, en un número multiplicado por una suma «A*(X+Y+Z)» el resultado es igual a la suma de los productos del número por cada uno de los sumandos «A*X+A*Y+A*Z». Pues dados dos números binarios, multiplicando que llamare-

de igual forma que la primera. Finalmente, se acaba con la tercera zona que está organizada de la misma forma que las dos anteriores. Esto es lo que nos permitía intercambiarlas (en la rutina para intercambiar zonas de pantalla) sin ningún problema; sin embargo, no es tan fácil intercambiar líneas, ya que sus scans no son consecutivos.

Si ha entendido a la perfección las explicaciones dadas hasta aquí sobre el archivo de pantalla, debe tener usted un coeficiente intelectual de super-dotado. Lo mejor, en cualquier caso, es que lo vuelva a leer detenidamente y, si es necesario, que se ayude de lápiz y papel para entenderlo mejor. La organización del archivo de pantalla no es nada fácil de comprender, pero es imprescindible manejarlo a la perfección para hacer rutinas de código máquina que trabajen sobre ella.

Pudiera parecer que con esta organización tan caótica, manejar la pantalla en código máquina es cosa de locos. Sin embargo, ahora veremos que resulta extremadamente fácil; es más, resulta más sencillo así que si los scans fueran consecutivos.

Ahora que, tras arduo esfuerzo, hemos sido capaces de comprender cómo está organizada la pantalla, vamos a olvidarnos de ello durante un momento. Lo que a nosotros nos interesa es ver de qué forma podemos hallar las direcciones de los 8 bytes que componen un determinado carácter, partiendo de sus coordenadas. Como ya sabe el lector, el archivo de pantalla ocupa los 6 primeros Kilobytes de la RAM, es decir,



Fig. 9-16. Formato de una dirección en el archivo de pantalla.

está colocado a partir de la dirección 16384 (4000h).

Por una curiosa «coincidencia» totalmente intencionada, podemos componer la dirección de cualquier octeto de la pantalla con sólo colocar de determinada forma los bits individuales de cada una de sus coordenadas. Vamos a verlo con sumo detalle porque es muy importante:

Dado que los números de línea y columna no pueden valer más de 23 y 31 respectivamente, cada uno de ellos sólo nos ocupa 5 bits. Supongamos que tenemos en «DE» las coordenadas de un determinado carácter, el registro «D» contendrá el número de línea que podrá estar comprendido entre 0 y 23, por tanto, sus tres bits de más a la izquierda serán «ceros». Llamaremos desde «l0» hasta «l4» a los cinco bits que nos indican la línea. De la misma forma, el registro «E» tendrá sus tres bits de la izquierda a «0» y llamaremos desde «c0» hasta «c4» a los cinco bits de la derecha que nos van a determinar la columna.

Queda un último dato: cada carácter tiene 8 octetos que pertenecen a 8 scans diferentes, y cada octeto tiene su dirección. Normalmente, nos interesará hallar la dirección del primero de estos octetos, pero no siempre así que será mejor añadir un dato más. Dado que se trata de 8 posibilidades, podemos representarlo con 3 bits, «000» será el octeto perteneciente al primer scan de la línea y «111» el que pertenezca al último scan de la misma. Vamos a llamarlos desde «s0» hasta «s2». Tenemos:

```
Número de linea:     0  0  0  14  13  12  11  10
Número de columna:   0  0  0  c4  c3  c2  c1  c0
Número de scan:                   s2  s1  s0
```

yos atributos estamos buscando, «D» deberá contener la línea y «E» la columna. A la salida de la rutina, tendremos, en «HL», la dirección del octeto del archivo de atributos, correspondiente a este carácter. El listado Assembler de «ATR» es el siguiente:

```
100 ATR   LD   A,D
110       SRA  A
120       SRA  A
130       SRA  A
140       ADD  A,#58
150       LD   H,A
160       LD   A,D
170       AND  #7
180       RRC  A
190       RRC  A
200       RRC  A
210       ADD  A,E
220       LD   L,A
230       RET
```

Vamos a ver cómo funciona con un ejemplo: Supongamos que queremos hallar la dirección de los atributos correspondientes al carácter cuyas coordenadas son (12,15), es decir, línea 12 (0Ch), columna 15 (0Fh). Veamos los pasos:

```
LD   A,D    ; #C -> A       A= 00001100 = #0C
SRA  A      ; A/2           A= 00000110 = #06
SRA  A      ; A/2           A= 00000011 = #03
SRA  A      ; A/2           A= 00000001 = #01
ADD  A,#58  ; A + #58       A= 01011001 = #59
LD   H,A    ; A -> H        H= 01011001 = #59
LD   A,D    ; D -> A        A= 00001100 = #0C
AND  #7     ; A and #7      A= 00000100 = #04
RRC  A      ; A/2           A= 00000010 = #02
RRC  A      ; A/2           A= 00000001 = #01
RRC  A      ; A/2           A= 10000000 = #80
ADD  A,E    ; A + E         A= 10001111 = #8F
LD   L,A    ; A -> L        L= 10001111 = #8F
RET
```

En el retorno, «HL» contiene el número # 598F, es decir, 22927 en decimal. Vemos que, efectivamente:

```
22528 + 12 x 32 + 15 = 22927
```

Por lo que 22927 es la dirección de los atributos correspondientes al carácter de coordenadas (12,15). Esta rutina trabaja perfectamente siempre que la línea esté comprendida entre 0 y 23; y la columna lo esté entre 0 y 31.

Antes de proseguir, conviene hacer referencia a la forma en que se almacenan los atributos, de un determinado carácter, dentro de cada byte del fichero de atributos. Empezando por la derecha, los tres primeros bytes almacenan el color de la tinta; los tres segundos, el color del papel; el séptimo, el flag de brillo y el octavo, el flag de parpadeo. Cuando la ULA va leyendo el fichero de pantalla para mandarlo al televisor, lee también los datos del fichero de atributos para mandar correctamente los colores de cada pixel.

Una vez vista la organización del fichero de atributos, vamos a ver la del fichero de pantalla. En este caso, las cosas no son tan sencillas como en el anterior. Sabemos que cada scan ocupa 32 bytes de memoria; parecería lógico que las direcciones de memoria de scans consecutivos fueran, también, consecutivas. Por desgracia, esto no es así.

Podemos considerar al archivo de pantalla como dividido en tres zonas de 2048 bytes cada una. La primera, correspondería a las 8 primeras líneas (0 a 7); la segunda, a las 8 segundas (8 a 15) y la tercera, a las 8 terceras (16 a 23). Para nuestros efectos, la pantalla consta de 24 líneas, así que podemos olvidarnos de que las dos líneas inferiores no son accesibles. Desde código máquina podremos acceder por igual a cualquier línea de la pantalla.

Ahora, vamos a estudiar la disposición de cada una de estas tres zonas, empezando por la primera. Cada línea se compone de 8 scans; pero no son consecutivos. Los primeros 32 bytes del archivo, contienen el primer scan de la primera línea (línea 0); los 32 siguientes, contienen el primer scan de la segunda línea; y así sucesivamente hasta el octavo grupo de 32 bytes que contiene el primer scan de la línea 7.

El noveno grupo de 32 bytes contiene el segundo scan de la primera línea, el décimo grupo, contiene el segundo scan de la segunda y así sucesivamente, hasta llegar al último grupo que contendrá el último scan de la octava línea (línea 7).

A partir de aquí, se comienza con la segunda zona de la pantalla que está organizada

mos MUL y multiplicador que llamaremos MOR, podemos descomponer el multiplicador en sus sumandos; y los sumandos pueden ser el valor relativo de cada bit; el resultado lo dejaremos en un campo llamado RES.

La técnica consiste en ir multiplicando MUL por cada sumando de MOR, que siempre será una potencia de dos, lo cual podemos hacer simplemente desplazando a la izquierda. Veamos un ejemplo:

```
30×26

30-11110 binario
26-11010 binario

11010-2^1+2^3+2^4-2+8+16

11110*2^1-   111100
11110*2^3-   11110000
11110*2^4-   111100000
suma total en RES 1100001100-780

Efectivamente 30×26-780.
```

El lector es posible que ya se diera cuenta que es la misma técnica que se sigue cuando se hace una operación de números de varias cifras en un papel, sólo que los desplazamientos se realizan inconscientemente.

EJEMPLO:

```
3456×752
     +1
   3456
   ×752
   6912
  17280      desplazamiento
 24192       desplazamiento
2598912
```

RECOMENDACIONES:

a) Si se opera con un solo octeto, ni los multiplicandos ni el resultado podrán superar 256 decimal.

b) Si se opera con números que ocupan más de un octeto, hay que tener en cuenta el bit que sale por la izquierda en los desplazamientos y el acarreo en las sumas; lo cual es posible ya que estas instrucciones (sumas y desplazamientos) dejan ambos en el indicador de acarreo «C».

c) Es muy conveniente poner límites fijos a los multiplicandos, en función de las necesidades. Lo más útil es que los dos tengan el mismo tamaño y el resultado la suma de los dos tamaños. De tal forma que si se decide que ambos ocupen un máximo de cuatro octetos el resultado estará dimensionado con ocho.

Ver Figura 9-14 que es el organigrama de una rutina de multiplicación de dos números MUL y MOR de 4 octetos de tamaño dejando el resultado en RES que tiene 8 octetos de tamaño. Los desplazamientos se señalan con una flecha indicando el sentido y un número indicando la cantidad. El campo RES se crea inicialmente a ceros. Tenga en cuenta que, tanto en las sumas sobre RES como en los desplazamientos de MUL y MOR, hay que manejar el indicador de acarreo «C» ya que los números están en más de un octeto.

En la división, podemos obtener soluciones parecidas así que vamos a analizar, paso a paso, el mecanismo, igual que con la multiplicación.

Recordemos que para dividir en decimal entre la unidad seguida de ceros o sea, potencias de diez, se corre una coma tantos lugares a la izquierda como ceros tenga la unidad, lo que es lo mismo, como valor tenga el exponente de la base, en este caso el 10. Esto es lo mismo que desplazar el número a la derecha, y el resto serían los números salientes justificados a la derecha. Ejemplo:

```
400:10

  4 0 0
|-|-|-|-|-|
    4 0
|-|-|-|-|-|

374:100

  3 7 4
|-|-|-|-|-|
      3
|-|-|-|-|-|
  resto
    7 4
|-|-|-|-|-|
```

Por el mismo motivo, si desplazamos un número binario una posición a la derecha, este número queda dividido por la base. Cada uno de sus bits pasarían de valer 2 elevado a «e», a valer 2 elevado a («e»-1).

```
0 1 0 1 0   -10
|-|-|-|-|-|
0 0 1 0 1    -5
|-|-|-|-|-|
```

Desplazarlo otra posición equivaldría a dividirlo otra vez por dos, esto es, dividir el número original por $4=2\times 2$, y así sucesivamente. Lo que se observa es que un nuevo desplazamiento implica dividir por la siguiente potencia de 2,

esto es 2, 4, 8, 16, 32, etc. Por lo tanto podríamos enunciar la siguiente regla: para dividir un número binario por una potencia de 2 se desplaza el número a la derecha tantas veces como valor tenga el exponente, y el resto serían los bits salientes por la derecha justificados a la derecha.

```
NUMERO__BINARIO : 2^E =
NUMERO__BINARIO → «E» veces

Ejemplos:

0 1 1 1 1 0 0 : 2^3 =
0 0 0 0 1 1 1 resto 1 0 0

60:8=7 resto 4

0 0 1 0 1 0 0 : 2^2 =
0 0 0 0 1 0 1 resto 0

20:4=5 resto 0
```

Al igual que en la multiplicación, no siempre se quiere multiplicar un número que sea una potencia de dos, es más, la mayoría de las veces no es así.

En un micro-procesador que carece de instrucción de dividir, una forma inmediata de solucionar el problema es restar el divisor al dividendo hasta llegar a cero o a un número menor que el divisor, entonces el cociente sería el número de restas. Ejemplo 40:10 habría que restar cuatro veces 10 hasta alcanzar el valor cero. Resultan pocas ejecuciones cuando la relación entre el dividendo y el divisor es pequeña y muchas cuando es grande.

Es necesario encontrar una técnica menos costosa, para lo cual empezaremos por observar cómo se realiza una operación de dividir números decimales de varias cifras en un papel:

Primero: Se toman las mínimas cifras de la izquierda del dividendo que sean iguales o mayores que el divisor, y a éstas les restamos «n» veces hasta llegar a cero o un número menor que el divisor, o lo que es lo mismo, buscamos un número «n» que multiplicado por el divisor valga igual o menos.

Segundo: Bajamos la cifra siguiente; o lo que es lo mismo, a lo que nos queda lo desplazamos una posición a la izquierda y metemos en el lugar entrante la cifra siguiente.

Y así repetimos la operación hasta que no tengamos nada que bajar o con que ocupar el lugar vacío.

```
47632 : 234
-468    203
---
 832
-702
---
 121

Cociente: 202
Resto: 121
```

Si lo hacemos con números binarios es mucho más fácil, pues siempre que alcancemos un valor superior al divisor, el número por el que tendremos que multiplicarlo será 1 y mientras no lo alcancemos se irán metiendo ceros en el cociente como en el ejemplo de división decimal.

La técnica es la siguiente: dado un dividendo DIV y un divisor DOR, pretendemos obtener un cociente COC y un resto RES.

a) Es necesaria la utilización de campos de tamaño fijo e igual.

b) El campo COC se definirá inicialmente a ceros.

c) Definiremos un campo DIV' de igual tamaño que DIV y pegado a él por la izquierda de forma que entre los dos DIV'__DIV formen un campo doble.

Y ahora, siguiendo los pasos parecidos a la división decimal desplazaremos a la izquierda DIV'__DIV hasta tener en DIV' un valor igual o mayor que DOR (divisor).

En este momento el número por el que hay que multiplicar el divisor siempre es 1 el cual se pone en el campo COC (cociente).

A partir de este punto se irán desplazando COC y DIV'__DIV hasta alcanzar el valor del divisor DOR, momento en que se restará de DIV' el divisor y se sumará uno al cociente.

Parece un lío pero pruebe con un ejemplo, nunca falla; primero uno sencillo, 1001010:110, y después uno más complicado como 1011011010101001:10110-01.

El resto será el valor que quede en DIV' cuando no haya más bits en el dividendo.

Ver organigrama de la Figura 9-11 hecho para campos fijos de 4 octetos con los números justificados a la derecha.

Los archivos de pantalla y atributos

Antes de estudiar cómo el Spectrum almacena, en memoria, la imagen que vemos en la pantalla, tal vez sea conveniente echar un vistazo a la forma en que se genera una imagen de televisión.

Como casi todos los lectores sabrán, la imagen de un televisor está compuesta por líneas horizontales. En la norma de televisión europea, se utilizan 625 líneas para formar un cuadro, aunque no todas entran en la composición de la imagen. El haz de electrones barre la imagen de izquierda a derecha y de arriba a abajo, por tanto, cuando termina de formar un cuadro, se encuentra en el extremo inferior derecho. Antes de formar el siguiente cuadro, el haz debe retornar al extremo superior izquierdo en la operación que se denomina «retorno de cuadro». Durante esta operación, se pierden algunas líneas, ya que el barrido horizontal continúa funcionando.

En el caso del Spectrum, aún se pierden unas cuantas líneas más en las zonas superior e inferior del «BORDER», así como parte de las restantes en los extremos derecho e izquierdo de la pantalla (el resto del «BORDER»). Al final, quedan 384 líneas en la parte de la pantalla que denominamos «PAPER». De estas líneas, el Spectrum sólo utiliza la mitad, es decir, una sí y una no. Por tanto, la imagen enviada por el Spectrum al televisor consta de 192 líneas.

A efectos de terminología, llamaremos «SCAN» a cada una de estas 192 líneas. Tendremos, por tanto, 192 «scans» en pantalla (la palabra «scan» significa, en inglés: «barrido»; utilizamos la terminología inglesa por ser la de uso más común en todo el mundo).

Cada scan de pantalla consta de 256 pixels. Un «pixel» es un elemento de imagen, un punto, que puede tener el color del papel o el de la tinta. Cuando hacemos un «PLOT» en Basic, ponemos un pixel del color de la tinta.

Cada pixel está controlado por un bit del archivo de pantalla de forma que, si el bit correspondiente es un «1», el pixel tendrá color de tinta y si es un «0», lo tendrá de papel.

Los colores de la tinta y el papel, así como los atributos de brillo y parpadeo, se encuentran en otra zona de memoria denominada «archivo de atributos». Llamamos «carácter» a un conjunto de 64 pixels, colocados en una matriz de 8×8. Por tanto, cada carácter ocupa 8 pixels de 8 scans consecutivos. Los 8 pixels de cada scan dentro de un carácter, están agrupados en una determinada posición de memoria, es decir, componen un octeto. Por tanto, cada carácter consta de 8 octetos; pero estos no son consecutivos. Por otro lado, cada elemento del archivo de atributos afecta a un carácter, por ello, hay 8 veces menos octetos en el archivo de atributos que en el de pantalla.

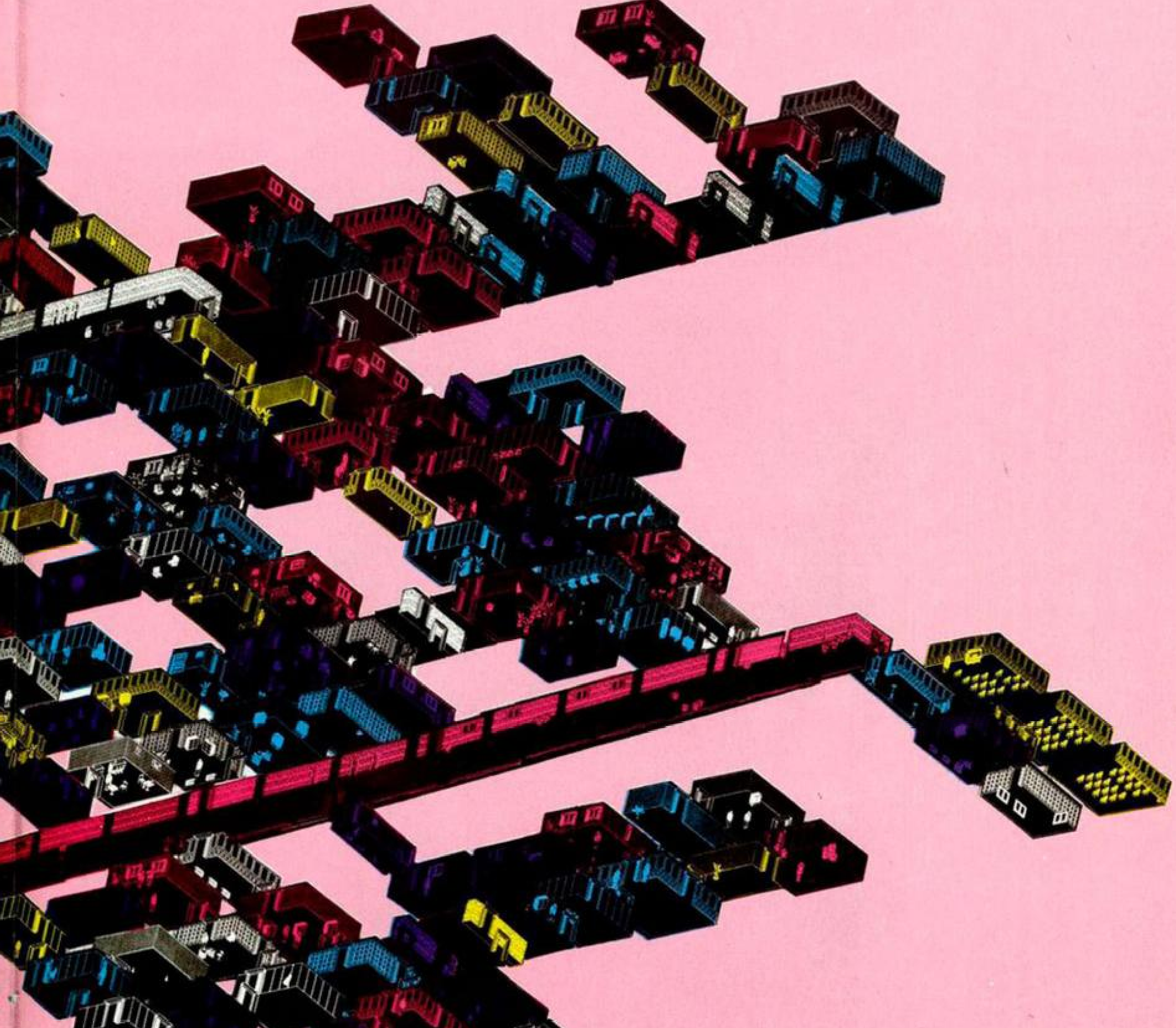
Si llamamos «línea» a un conjunto de 32 caracteres colocados uno al lado del otro, y «columna» a un conjunto de 24 caracteres colocados uno encima del otro, podemos decir que la pantalla del Spectrum tiene 24 líneas y 32 columnas, es decir, 24×32=768 caracteres. Por tanto, el archivo de atributos tendrá 768 bytes (u octetos) y el de pantalla, 768×8=6144 bytes. Si al llegar a este punto, no ha comprendido algo de lo anterior, vuelva a leerlo más despacio antes de proseguir.

Vamos a empezar por describir cómo está colocado el archivo de atributos ya que es el más fácil, luego, pasaremos al de pantalla que es bastante más complejo.

Los 768 bytes colocados a partir de la dirección de memoria 22525 (5800h) inclusive, constituyen el archivo de atributos. El primer byte corresponde al carácter del ángulo superior izquierdo (coordenada 0,0). El segundo, al siguiente de esa línea (coordenada 0,1) y así sucesivamente hasta el último que corresponde al carácter del ángulo inferior derecho (coordenada 23,31; aunque esta coordenada no existe en Basic, ya que corresponde a las dos líneas inferiores que, como se sabe, son usadas por el canal «K»).

La finalidad de todo esto es ser capaces de averiguar la dirección del atributo de un determinado carácter, conociendo sus coordenadas. En este caso, el problema tiene fácil solución: bastará con multiplicar el número de línea por 32, sumarle el número de columna y, al resultado, sumarle 22528 que es la dirección base de este archivo. En realidad, no difiere mucho de una tabla indexada con dos subíndices o, si lo prefiere, una matriz de dos dimensiones.

Para hacer esto, en código máquina, tendremos en cuenta que multiplicar un número por 32 equivale a desplazarlo 5 lugares a la izquierda. También podemos usar la rutina de multiplicar publicada en este mismo curso, aunque es más lenta. Si, después de todo, decide hacerlo por rotaciones, tal vez se encuentre con el problema de que, al rotar a la izquierda el número de línea, se le escapará por el lado izquierdo del registro. Afortunadamente, existe una rutina que lo hace de forma sencilla. Hemos denominado «ATR» a la rutina, ya que sirve para buscar un atributo determinado. En ella se entra con «DE» conteniendo las coordenadas del carácter cu-





Imagine
...the name of the game

Profesor particular



Arturo Lobo y J. J. León

DESPEJAR

Como bien indica su título, el programa de esta semana permite despejar la «x» de cualquier expresión, de cualquier ecuación, que la introduzcáis, siempre y cuando la «x» sólo aparezca una vez en la expresión.

El programa es principalmente de carácter didáctico y está especialmente indicado para aquellos lectores en cursos altos de EGB o en los cursos de BUP y, en general, para los estudiantes que no tienen mucha soltura despejando.

No nos cabe duda que con el programa y un poco de trabajo podrán solucionar todas sus dudas y adquirir mayor destreza en este tema.

Entre las posibilidades del programa cuenta con una que os plantea un problema de despejar la dificultad que elijáis.

Además podéis contar con una serie de ayudas con los paréntesis o con las operaciones y funciones que os indican a cada paso cuáles son las funciones u operaciones que se deben pasar al otro miembro. Aunque lo más indicado es que vayáis siempre un paso por delante del ordenador y confirméis después que lo habéis hecho bien.

Siempre es conveniente ir escribiendo u obteniendo las expresiones sobre el papel pues la anotación de paréntesis de un ordenador no siempre es clara.

La ayuda de paréntesis lo que hace es imprimir en INVERSE las partes de la expresión que están dentro de paréntesis y como la operación o función a despejar ha de estar fuera de estos paréntesis, será más fácil de encontrar.

La ayuda de «operaciones» imprime en Flash 1 la operación o función que se debe despejar.

Cuando se introduce una expresión para despejar hay que tener cuidado de no olvidar ningún paréntesis ni el igual o la «x», si no el programa avisará de ello. Como hemos dicho, la «x» sólo puede aparecer una vez en la expresión, si aparece más veces sólo toma como incógnita la primera que encuentra.

Cuando se han solicitado las ayudas, el programa pide permiso; para continuar basta con pulsar una tecla para que lo haga.

Cuando ha llegado al final y tiene despejada la «x», se le puede pedir que obtenga la solución numérica, sin embargo, hay que tener en cuenta varias cosas: puede dar error, si tiene que calcular arcoseno o arcosenos de expresiones mayores que 1.

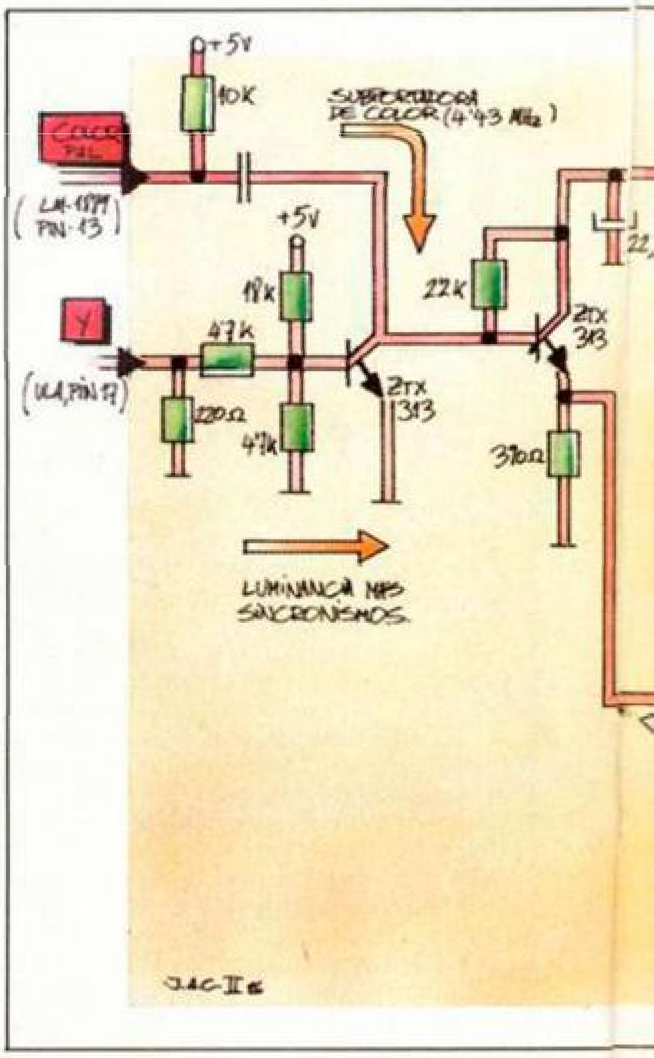
Además, siempre que aparezca la operación elevar, el programa la despeja poniendo valores absolutos a las expresiones que intervengan ya que en el Spectrum no se puede realizar (—2)↑2 por ejemplo, o sea, que podría llegar a una solución errónea.

```
1999 DEF FN r(x)=asd+INT (RND*x)
2000 BORDER 6: PAPER 6: INK 0: CLS : LET asd=1: LET i$="x"
2020 INPUT "QUIERES QUE TE PLANTEE UN EJER- CICIO (s/ENTER) "; LINE J$
2030 IF J$<>"" THEN GO SUB 9600: GO TO 2050
2040 INPUT "QUE EXPRESION QUIERES QUE DESPE-JE : "; LINE E$
2050 PRINT AT 0,0;e$: BEEP .5,20: PRINT AT 16,0;"AYUDAS :";"P- parentesis";"O- operaciones";"N- ninguna";"A- ambas dos"
2060 LET j$=INKEY$: IF j$<>"" THEN LET en1=(j$="p")+(j$="a"): LET en2=(j$="o")+(j$="a"): GO TO 2070
2065 GO TO 2060
2070 CLS : PRINT AT 0,0;e$
2080 GO SUB 9000: PRINT AT 17,0;"SOLUCION : ";I$;"=";B$
2090 INPUT "QUIERES LA SOLUCION NUMERICA ";"(s/n) "; LINE j$
2100 IF j$="s" THEN BEEP .5,10: PRINT AT 15,0;i$;"=";VAL b$
2110 STOP
9000 LET d$=e$: GO SUB 9480: LET pi=pvar
9010 FOR i=1 TO LEN E$: IF E$(i)="=" THEN LET pe=i: GO TO 9020
9015 NEXT i: PRINT "NO HAY ""=""!!!": RETURN
9020 IF pe>pi THEN LET B$=E$((pe+1) TO LEN E$): LET A$=E$(1 TO (pe-1)): GO TO 9030
9025 LET B$=E$(1 TO (pe-1)): LET A$=E$((pe+1) TO LEN E$)
9030 LET d$=a$: GO SUB 9480: LET pi=pvar
9035 IF A$=I$ THEN : RETURN
9040 LET cuenta=0: DIM a(25,2): DIM b(25): LET ax=1: LET np=0: LET a(1,2)=0: FOR i=1 TO LEN A$: IF CODE A$(i)=40 THEN LET cuenta=cuenta+1: LET np=np+1: IF np=1 THEN LET ax=ax+1: LET a(ax,1)=i: NEXT i
9045 IF CODE A$(i)=41 THEN LET np=np-1: LET cuenta=cuenta-1: IF np=0 THEN LET a(ax,2)=i: NEXT i
9047 NEXT i: IF cuenta<>0 THEN PRINT "numero impar de parentesis !!!": RETURN
9050 LET a(ax+1,1)=LEN A$+1: LET a(ax+1,2)=LEN a$+1: LET a(1,1)=0: IF a(2,1)=1 THEN IF a(2,2)=LEN A$ THEN LET A$=A$(2 TO (LEN A$-1)): LET pi=pi-1: IF a(ax,2)<>0 THEN GO TO 9035
9052 IF (en1=1)*(ax>1)=1 THEN GO SUB 9550
9055 LET no=0: LET niv=0: LET b(1)=0: FOR j=1 TO ax: FOR i=a(j,2)+1 TO a(j+1,1)-1: LET orden=FN p(CODE A$(i)): IF orden>niv THEN LET niv=orden: LET no=2: LET b(2)=i: GO TO 9062
9060 IF orden=niv THEN IF niv>0 THEN LET no=no+1: LET b(no)=i
9062 NEXT i: NEXT j: LET b(no+1)=LEN A$+1
9065 IF EN2=1 THEN GO SUB 9560+10*(NIV<>0)
9070 GO SUB (9100+100*niv): BEEP .2,30: PRINT AT 5,0;A$;"=";B$;" ": GO TO 9036
9080 DEF FN P(x)=3*((x=43)+(x=45))+2*((x=42)+(x=47))+(x=94)
9100 IF CODE A$(1)=187 THEN LET A$=A$(2 TO LEN A$): LET B$="ABS ("+B$+")↑2": RETURN
9105 DEF FN I(x)=189*(x=189)+45*(x=43)+43*(x=45)+178*(x=181)+181*(x=178)+179*(x=182)+182*(x=179)+180*(x=183)+183*(x=180)+184*(x=185)+185*(x=184)+42*(x=47)+47*(x=42)
9110 LET B$=CHR$ FN I(CODE A$(1))+"("+B$+")": LET A$=A$(2 TO LEN A$)
9115 IF FN I(CODE A$(1))>0 THEN IF en2=1 THEN GO SUB 9560
9120 IF FN i(CODE a$(1))>0 THEN LET B$=CHR$ FN I(CODE A$(1))+B$: LET A$=A$(2 TO LEN A$): GO TO 9115
9125 RETURN
9200 FOR i=1 TO no: IF b(i)<pi THEN LET posx=i
9205 NEXT i
9210 IF posx=1 THEN LET C$=A$(b(2)+1 TO b(3)-1): FOR i=3 TO no: LET C$=C$+"*"+A$(b(i)+1 TO b(i+1)-1): NEXT i: LET B$="ABS ("+B$+")↑(1/("+C$+"))": LET A$=A$(1 TO b(2)-1): RETURN
9220 LET C$="ABS "+A$(1 TO b(2)-1): FOR i=2 TO no: IF i<>posx THEN LET C$=C$+"↑"+A$(b(i)+1 TO b(i+1)-1)
9230 NEXT i: LET B$="LN ABS ("+B$+")/LN ABS ("+C$+")": LET A$=A$(b(posx)+1 TO b(posx+1)-1): RETURN
9300 GO TO 9400
9400 LET B$="("+B$+")": IF b(2)>1 THEN LET A$=CHR$ (42+(niv=3))+A$: FOR i=1 TO no+1: LET b(i)=b(i)+1: NEXT i: LET pi=pi+1: GO TO 9410
9405 FOR i=1 TO no: LET b(i)=b(i+1): NEXT i: LET no=no-1
9410 FOR i=1 TO no: IF b(i)<pi THEN LET posx=i
9415 NEXT i
9420 IF A$(b(posx))=CHR$ (42+(niv=3)) THEN GO TO 9440
9425 LET C$="": FOR i=1 TO no: IF i<>posx THEN LET C$=C$+A$(b(i) TO b(i+1)-1)
9430 NEXT i: LET C$=C$(2 TO LEN C$): LET B$=C$+CHR$ (45+2*(niv=2))+B$: LET A$=A$(b(posx)+1 TO b(posx+1)-1): RETURN
9440 FOR i=1 TO no: IF i<>posx THEN LET B$=B$+CHR$ FN I(CODE A$(b(i)))+A$(b(i)+1 TO b(i+1)-1)
9445 NEXT i: LET A$=A$(b(posx)+1 TO b(posx+1)-1): RETURN
9480 IF d$="" THEN LET pvar=0: RETURN
9490 LET pvar=0: LET fin=LEN d$-LEN i$+1: FOR p=1 TO fin: IF d$(p TO (p-1+LEN i$))=i$ THEN IF ((FN l(CODE d$(p-1+(p=1)))=0)+(p=1))*((FN l(CODE d$(p+LEN i$-(p=fin)))=0)+(p=fin))=1 THEN LET pvar=p: RETURN
9492 NEXT p: RETURN
9498 DEF FN l(x)=x*((x>64)*(x<91)+(x>96)*(x<123)+(x>47)*(x<58))
9550 BEEP .5,50: PRINT AT 10,0: FOR k=1 TO ax: PRINT a$(a(k,2)+1 TO a(k+1,1)-1);: IF k<>ax THEN PRINT INVERSE 1;a$(a(k+1,1) TO a(k+1,2));: NEXT k
9555 GO TO 9590
9560 BEEP .4,20: GO SUB 9580: PRINT AT 10,0; FLASH 1;a$(1); FLASH 0;a$(2 TO LEN a$);
9565 GO TO 9590
9570 BEEP .3,0: GO SUB 9580: PRINT AT 10,0: FOR k=1 TO no: PRINT a$(b(k)+1 TO b(k+1)-1);: IF k<>no THEN PRINT FLASH 1;a$(b(k+1));: NEXT k
9575 GO TO 9590
9580 PRINT AT 10,0;: FOR g=1 TO 4*32: PRINT " ";: NEXT g: RETURN
9590 PRINT "=";b$: BEEP .2,20: BEEP .1,0: PRINT AT 20,20;"CONTINUO ?"
9592 IF INKEY$<>"" THEN PRINT AT 20,20;"          ": RETURN
9594 GO TO 9592
9600 LET yai=0: LET par=0: LET yax=0: LET r$=""
9605 INPUT "DIFICULTAD de 1 a 10 ";ldif
9610 LET ldif=3*ldif: GO TO 9670
9640 IF (FN r(12)>3+2*ldif/LEN r$+(par>(ldif>3))+30*yax)=1 THEN LET r$=r$+"x": LET yax=1: GO TO 9642
9641 LET r$=r$+CHR$ (49+FN r(8))
9643 LET rx=FN r(10): IF (par>0)*(rx>3+2*ldif/LEN r$) THEN GO TO 9650
9644 GO TO 9660+20*(pAr=0)*yax*(ldif/LEN r$<rx/(3+3*yai))
9650 LET par=par-1: LET r$=r$+")": LET rx=FN r(10)
9652 IF par=0 THEN GO TO 9660+20*yax*(2*FN r(ldif/LEN r$)<3)
9654 GO TO 9660-10*(FN r(ldif/LEN r$)<2)
9660 LET qw=FN r(5): LET qw=42*(qw=1)+43*(qw=2)+45*(qw=3)+47*(qw=4)+94*(qw=5): LET r$=r$+CHR$ qw
9662 GO TO 9670-30*(FN r(10*ldif/LEN r$)<3)
9670 IF FN r(12)>3+3*par+4*yax THEN LET par=par+1: LET r$=r$+"(": GO TO 9672
9671 LET r$=r$+CHR$ (177+FN r(8))
9672 IF FN r(20)>6+(3-yax+par)*(ldif/(3+1.5*LEN r$)) THEN GO TO 9640
9674 GO TO 9670
9680 IF yai=0 THEN LET yai=1: LET r$=r$+"=": GO TO 9670
9685 LET e$=r$: RETURN
```

EL MODULADOR DE VIDEO

Primitivo de FRANCISCO

Quien se haya atrevido a abrir su ordenador habrá podido observar en uno de sus ángulos una caja metálica con un conector en el que se inserta el cable apantallado que va directamente a la entrada de antena del televisor. Esta cajita contiene el modulador con cuya ayuda se introducen las imágenes vía radiofrecuencia en el receptor de TV.



El modulador es en sí mismo un «todo» capaz de añadir a la señal de vídeo una portadora de muy alta frecuencia para entrar en el televisor por la antena como si de una emisora más se tratara. Y, en efecto, el modulador es una diminuta emisora de TV de muy baja potencia capaz de radiar hacia el éter. Este efecto se puede apreciar si se desconecta la clavija del modulador y se mantiene a unos centímetros del mismo, en el televisor sigue percibiéndose la imagen, aunque con notable pérdida de calidad debida a la atenuación de la señal que se produce en el desacoplamiento entre el emisor y receptor. Con una antena correctamente colocada en cada uno de los dos (Modulador y Televisor) podrían ser enlazados a varios metros de distancia, el ordenador y el receptor de TV. Cuando ambos aparatos se hallan correctamente interconectados el efecto de radiación es mínimo, pues los apantallamientos del modulador y del cable hacia la antena están pensados para que su señal no perturbe a ningún otro receptor próximo.

En su día, en una Convención Internacional de Telecomunicaciones, se acordó dejar un espacio libre en la banda de UHF para pequeños aparatos que precisasen las prestaciones de un receptor de TV como son los videograbadores, los videojuegos y los ordenadores domésticos.

La banda escogida para este fin va desde el canal 32 al 36 cuyas frecuencias abarcan desde 558 hasta 597 Megaherzios. (La banda de UHF destinada a televisión comercial cubre desde 470 Mhz para el canal 21, hasta 853 Mhz para el canal 68). Estas frecuencias corresponden a las portadoras de las emisoras de televisión sobre las cuales «Cabalgan»: La señal de vídeo que modula en amplitud a la portadora, la señal de color que modula en fase una subportadora de 4,43 Mhz y la señal de audio que modula en frecuencia a otra subportadora de 5,5 Mhz (Sistema PAL).

El tratamiento de frecuencias de portadoras tan altas exige de ciertas medidas en la construcción y ajuste del modulador, como bien saben los conocedores del mundo de la radio, que encarecen verdaderamente el producto. Estas medidas hacen que se diferencien notablemente los dispositivos digitales a los analógicos y de radio. Cuando los técnicos de Sinclair decidieron incorporar este modulador a sus ordenadores ZX-Spectrum lo hicieron en función de una cierta economía. Echando un vistazo al catálogo de moduladores fabricados por ASTEC se observa que tiene otros muchos, incluso los que incorporan la señal del sonido por vía antena; pero éstos son de mayor tamaño y costo. Por otra parte, añadir también la modulación del sonido exige aún mayores cuidados en la realización del modulador para que éste no afecte a la señal de vídeo y en especial al color. El efecto negativo de la inclusión del sonido en el modulador, si éste no es de una cierta calidad, es la producción de franjas horizontales y distorsiones de color que fluctúan en la pantalla desagradablemente al ritmo de la señal de audio. Así pues, el modulador UM-1233 de ASTEC montado en el Spectrum no modula el sonido, tan sólo proporciona una modulación negativa de la señal de vídeo con una calidad media.

Circuitos asociados al modulador

En la figura n.° 1 se puede apreciar el entorno del modulador en el Spectrum. La ULA genera las tres señales básicas de vídeo: B-Y y R-Y información de color e Y que contiene el nivel de luminancia y sincronismos de línea y cuadro. Las señales B-Y y R-Y sirven para que el circuito LM-1889 proporcione la subportadora de color según la normativa PAL. La señal procedente del circuito integrado LM-1889 se suma con la señal Y de la ULA. A la salida del sumador, que está hecho con dos transistores, se obtiene la señal compuesta de vídeo conteniendo la totalidad de la información lista para ser presentada por pantalla, línea a línea y cuadro a cuadro, en un monitor de vídeo o vía modulador en un aparato de TV. El modulador tiene dos entradas además de la masa que va directamente a la carcasa metálica de su caja. Una de estas entradas es la alimentación del modulador (+5V), y la otra es la entrada de la señal de vídeo con una amplitud aproximada de 1 voltio pico a pico.

En el interior del modulador la señal

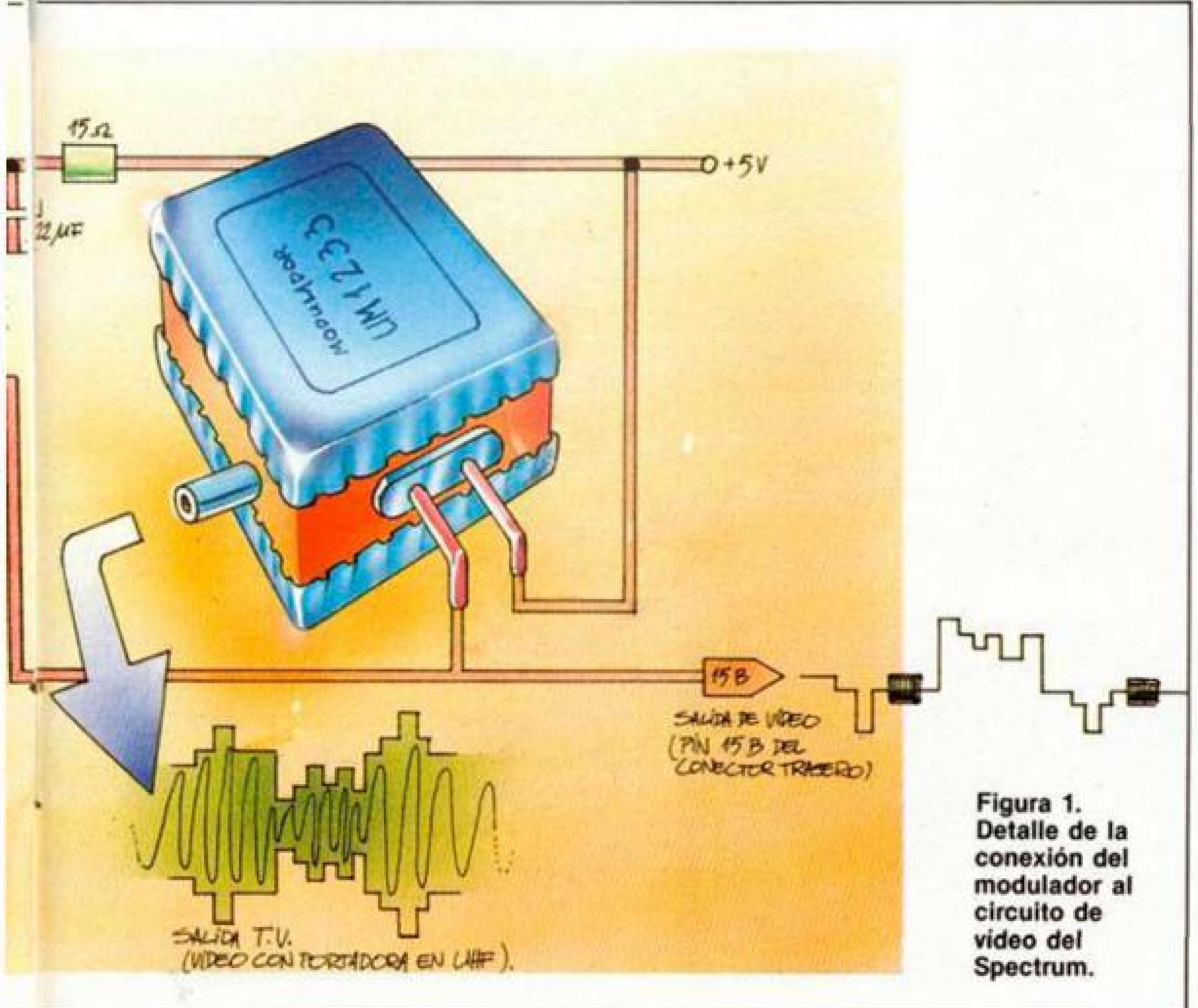


Figura 1. Detalle de la conexión del modulador al circuito de vídeo del Spectrum.

de vídeo modula a la portadora que es una oscilación de 591,5 Mhz (canal 36). La salida la hace por el conector coaxial hembra existente en uno de sus laterales. La impedancia típica de esta salida es de 75 ohm., idéntica a la del cable apantallado y a la de entrada de antena del televisor.

El modulador por dentro

El modulador está compuesto de dos etapas, una está formada por el circuito oscilador y la otra es la etapa moduladora.

El oscilador lo constituye un único transistor (T3) el cual oscila con el apoyo de unos cuantos componentes pasivos. La frecuencia viene determinada por los valores L-C del tanque compuesto por C7 y L5.

A la salida del oscilador la amplitud es constante y su frecuencia puede ser variada dentro de unos límites retocando el núcleo de ferrita que se halla en el interior de la bobina del tanque oscilador.

En la tapa superior del modulador existe un agujero para facilitar el acceso al núcleo de la bobina y retocar así la frecuencia del oscilador a un punto en que se eliminen posibles perturbaciones o interferencias, tanto con aparatos muy próximos como con posibles armónicos de emisoras o repetidores de TV cercanos. No se daña en modo alguno al ordenador por manipular esta bobina, así que hazlo si tienes necesidad de ello o por mera curiosidad.

La salida del oscilador pasa por R6, R7, R8 y C8 que conforman una red adaptadora para atacar al devanado derecho de la bobina L4 (ver figura n.º 2).

Los transistores T1 y T2 constituyen los elementos activos del circuito modulador. La señal de vídeo, que lleva una componente continua próxima a un voltio, polariza y controla por la base a ambos transistores. T1 y T2 trabajan en contrafase para obtener una mayor profundidad de modulación sin necesidad de excesivos componentes pasivos.

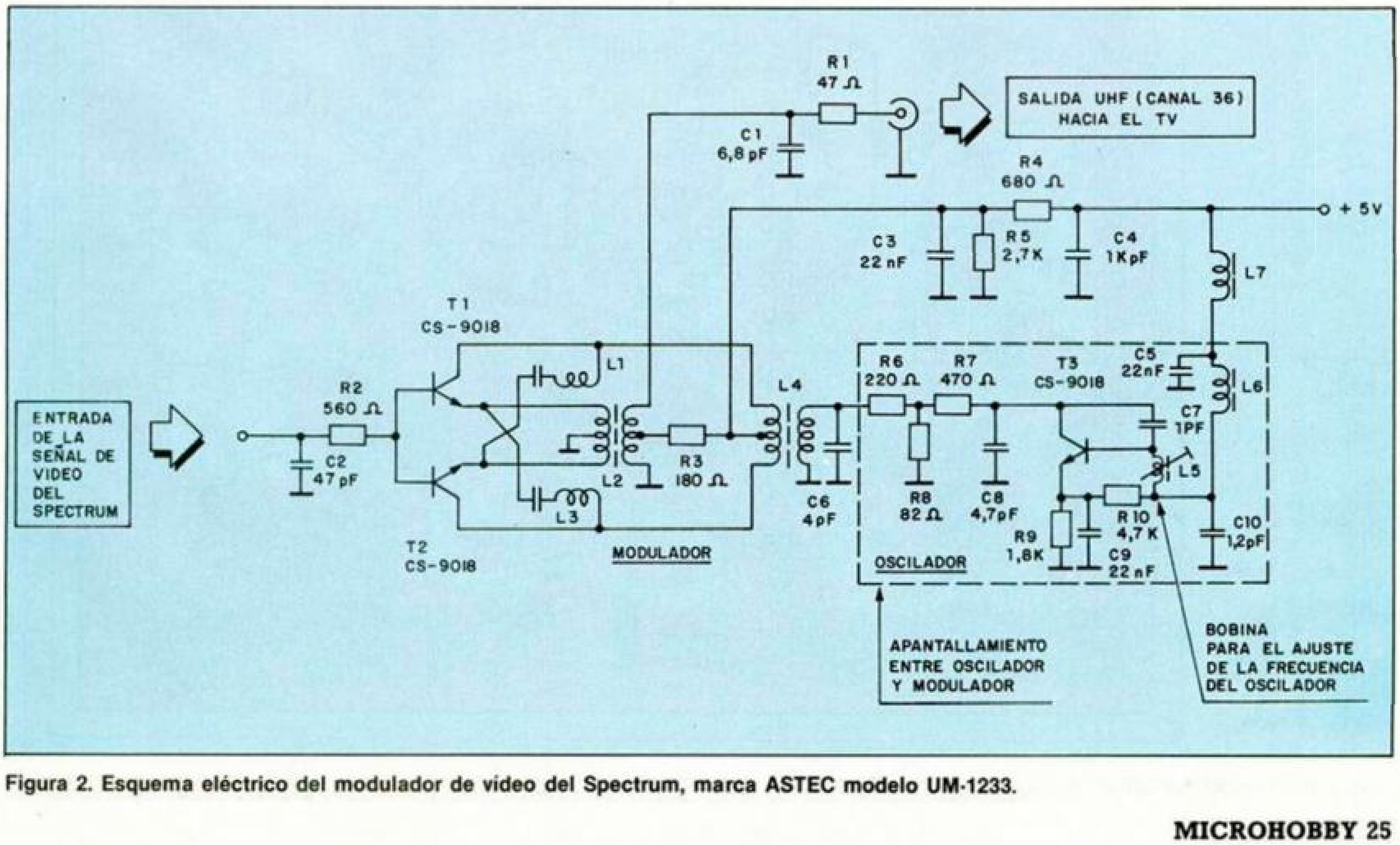
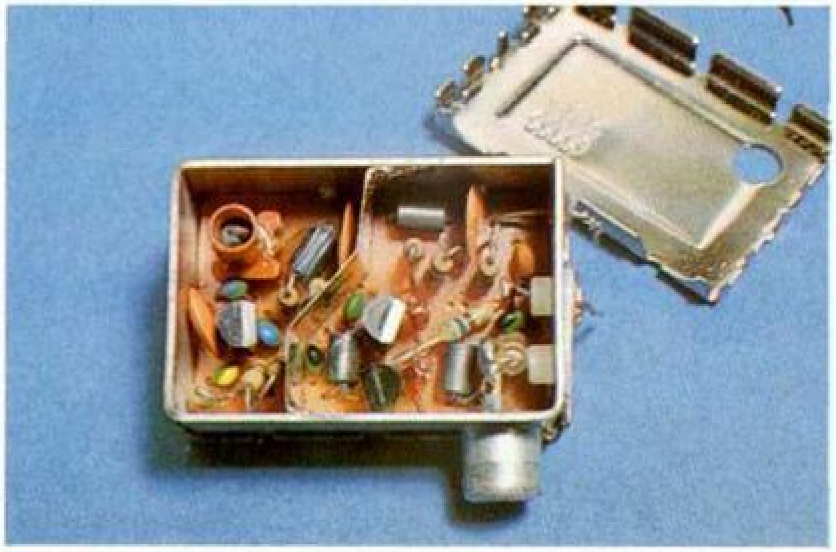
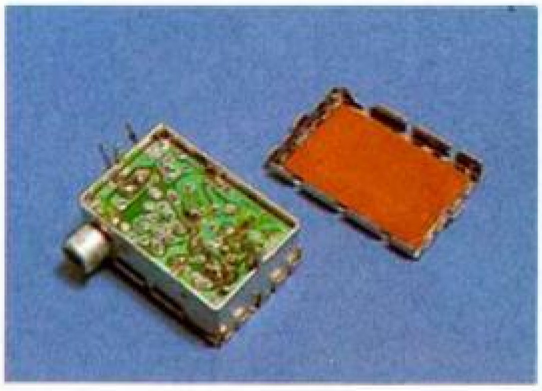


Figura 2. Esquema eléctrico del modulador de vídeo del Spectrum, marca ASTEC modelo UM-1233.



Vista superior del modulador una vez abierto, donde se observa la separación física entre el oscilador y el modulador.

De esta forma, la frecuencia de amplitud constante que entrega el oscilador es modulada por los transistores T1 y T2 con el auxilio de los transformadores L4 y L1; desde este último sale ya la señal en UHF perfectamente modulada. L4 y L1 no son ajustables, están compuestos de un núcleo de ferrita en donde están devanados directamente el primario y el secundario, compuesto por una o dos espiras (a tan altas frecuencias éstas proporcionan la inducción necesaria).



La aparente arbitrariedad en la colocación de los componentes se debe a las altas frecuencias que utiliza.

L6 y L7 son meros choques para evitar que la señal del oscilador se pierda por el circuito de alimentación. En el interior del modulador una lámina metálica separa físicamente al oscilador del modulador para evitar interferencias mutuas.

La figura n.° 3 muestra la disposición interna de los componentes del esquema eléctrico. Obsérvese la aparente arbitrariedad de su colocación, ello obedece a necesidades de realización frente a frecuencias tan altas aún a pesar de un detrimento estético. Obsérvese también, cómo L5 está confeccionada con media espira, a 591,5 Mhz, la L que presenta es suficiente. La alimentación del modulador es de 5 volt. y la salida de 75 ohm. para conectar con esta misma impedancia el televisor doméstico. Existen antiguos receptores de TV con una impedancia de entrada de 300 ohm. Para estos últimos es aconsejable incluir entre el ordenador y el televisor un adaptador de impedancia, fácil de encontrar en el comercio electrónico.

Esta explicación teórica será de gran ayuda para todos los que decidáis abordar un próximo montaje que tendrá por objeto mejorar las señales de vídeo y la calidad del color en el Spectrum.

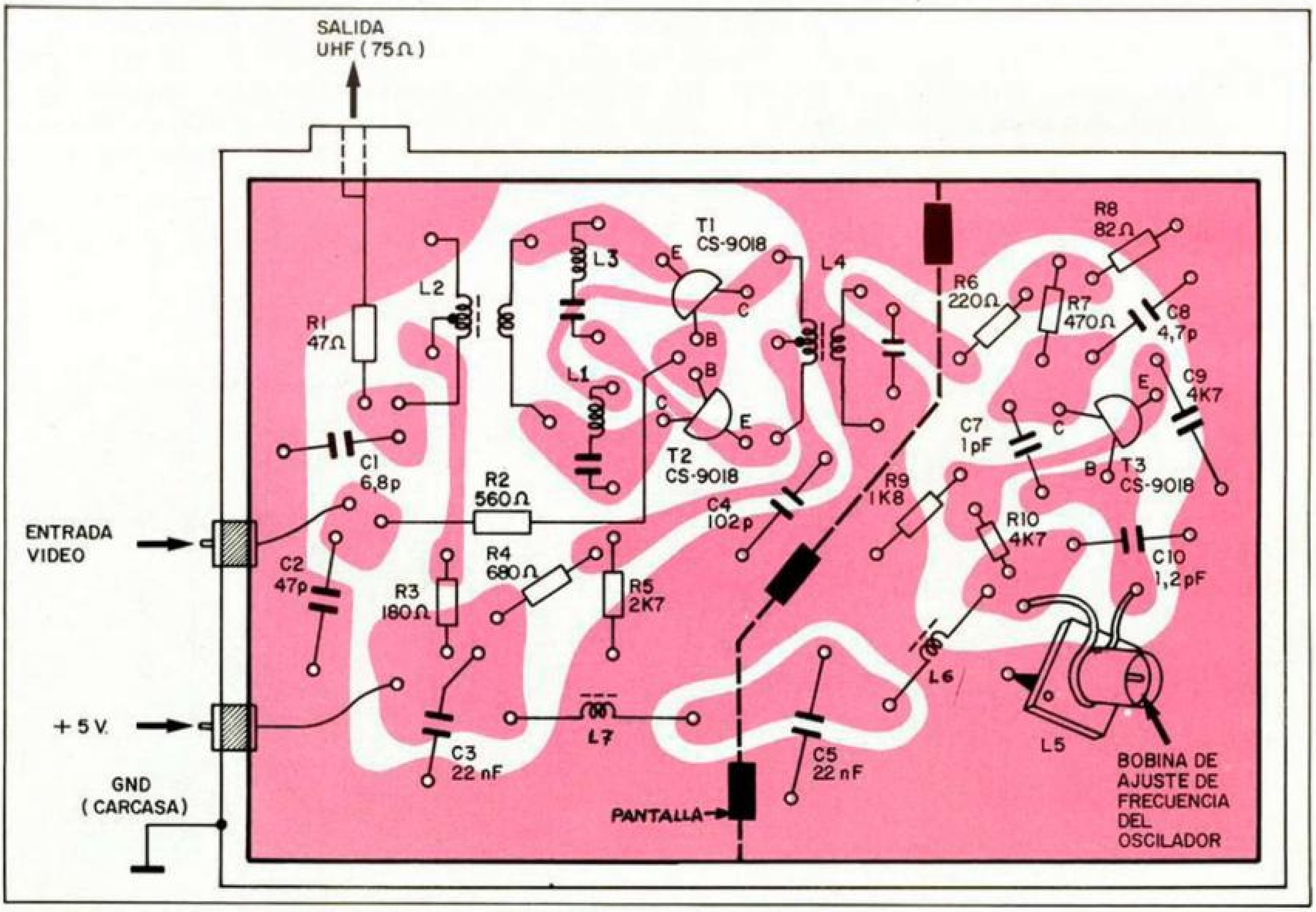


Figura 3. Interior del modulador de UHF de Astec UM-1233-E36 (canal 36, 591'5 MHz).

MICROHOBBY
PRESENTA

El mayor éxito del año en todas las pantallas de España

NOMINADA PARA SEIS OSCARS



MEJOR PROGRAMA

Ensamblador

MEJOR GUION

Editext

MEJOR DOBLAJE

Monitor

MEJOR GRAFISTA

Screen

MEJOR MUSICA

Data Beep

MEJOR MONTAJE

Renumerador

CON EL MAYOR DESPLIEGUE DE EXTRAS DE LA HISTORIA DEL SOFT

Desensamblador, Cargador CM., Depurador, Cirujano, Tokens, Micro CAT, Conversor, Listador, Copiupi, Audio Agenda, Multi UDG, Traspa, Voz.

Proyección especial hasta el 31 de mayo para todos nuestros lectores consistente en:

«2 CINTAS que contienen 20 PROGRAMAS DE UTILIDADES valoradas en 2.500 ptas.), gratis al realizar tu suscripción»

Recorta o copia este cupón y envíalo a Hobby Press, S. A., Apartado de Correos 232. Alcobendas (Madrid).

Nombre ______________ Apellidos ______________

Dirección ______________ C. Postal ______________

Localidad ______________ Provincia ______________

Teléfono ______________ Profesión ______________

Deseo suscribirme a **MICROHOBBY SEMANAL** (50 números) al precio de 5.670 ptas. (IVA incluido). Esta suscripción me da derecho a recibir **totalmente gratis, 2 cintas,** que contienen 20 programas de utilidades, valoradas en **2.500 ptas.** (Oferta válida hasta el 31 de mayo de 1986).

Deseo recibir en mi domicilio la cinta «20 utilidades», al precio de 2.500 ptas. (suscriptores 1.900 ptas.).
Número de suscriptor ______________ (si no lo recuerda escriba sólo la palabra «SI»).

FORMA DE PAGO. MARCA CON UNA X LA OPCION QUE DESEES.

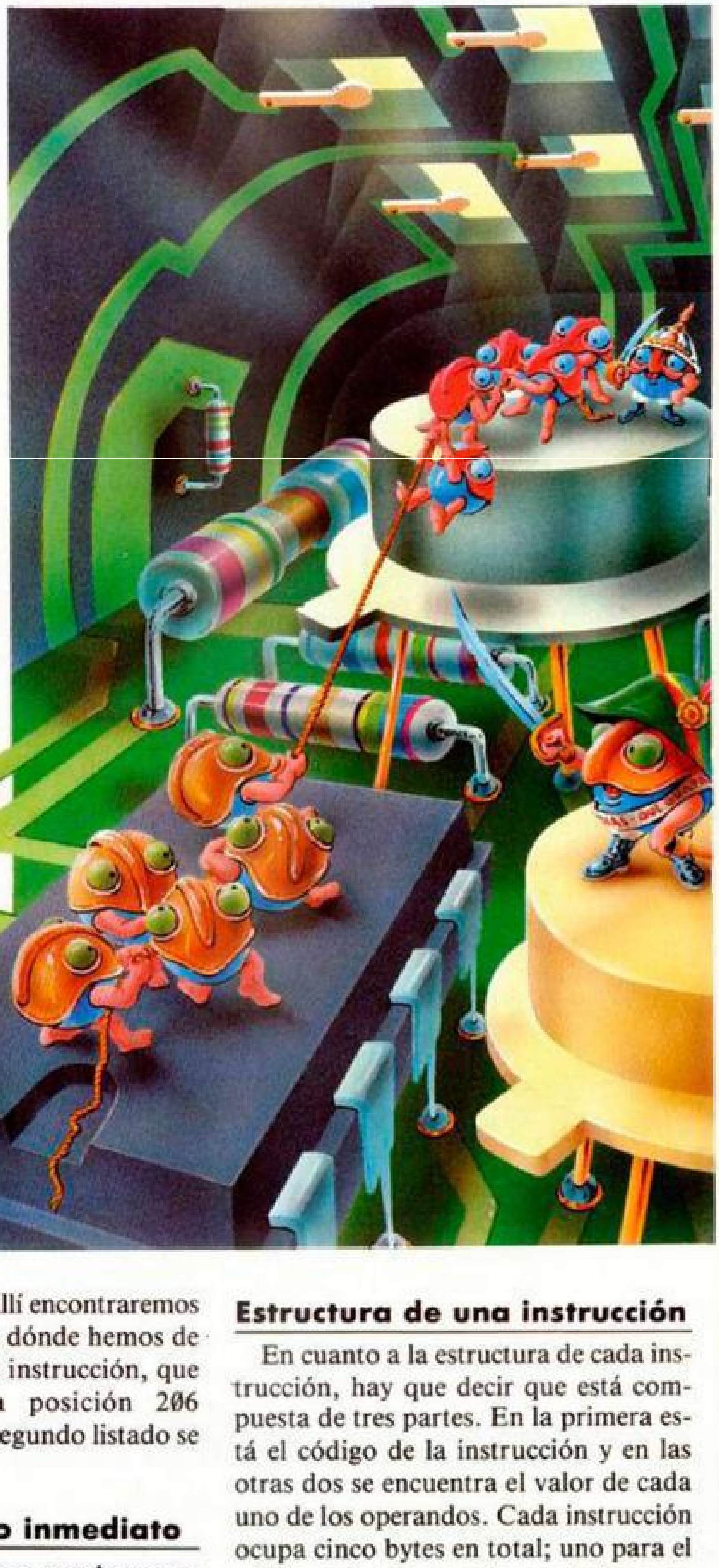
- ☐ Contra reembolso
- ☐ Mediante tarjeta VISA. Núm. de la tarjeta ______________ Fecha caduc. de la tarjeta ______________
- ☐ Mediante talón bancario a nombre de Hobby Press, S. A.
- ☐ Mediante giro postal n.° ______________
- ☐ Mediante domiciliación bancaria

Banco ______________ Sucursal y Localidad ______________

N.° de cuenta ______________ Fecha y firma

BATCODE (III)

Sergio MARTINEZ LARA



Vamos a describir a continuación las características generales del lenguaje BATCODE, destacando de entre ellas los modos de direccionamiento, una de sus facetas más complicadas.

El cálculo de las direcciones dentro del hipotético campo de batalla, se realiza siempre de forma relativa a la posición en la que nos encontramos en cada momento. Por ejemplo, observemos los listados de la figura 3; el de la izquierda es anterior a la ejecución y el de la derecha posterior a la misma. Para que se entienda el ejemplo adelantaremos que la instrucción MOV sirve para copiar instrucciones o datos de un lado a otro de la memoria.

```
202 SUM   3  2  202=203-1   |  202 SUM   3  2
203 MOV  -1  3              |  203 MOV  -1  3
... ... ... ..              |  ... ... ... ..
... ... ... ..              |  ... ... ... ..
206 ... ... ..              |  206 SUM   3  2  206=203+3
```

Figura 3. Direccionamiento relativo directo.

Direccionamiento directo

Dicho esto, supongamos que se ejecuta la instrucción 2Ø3, esto significa que se cogerá lo que se encuentre en la posición anterior (2Ø2) y se colocará tres posiciones más adelante (2Ø6 = 2Ø3 + 3). El resultado puede apreciarse en el listado de la derecha. No es motivo de preocupación el que de momento no se comprenda bien. Lo que se pretende ahora es ver cómo funciona el tema de las direcciones relativas. Este direccionamiento que hemos visto es el denominado modo «normal» o «directo».

Direccionamiento indirecto

Si la dirección tiene delante el signo (@) estamos en el modo «indirecto».

Consiste en tomar como dirección efectiva, no ya el valor del argumento, sino el valor de la posición de memoria indicada por dicho argumento. Esto se entenderá mejor con el ejemplo de la figura 4. En el primer listado de dicha figura se ve que primero se calcula la dirección relativa —2 y allí encontraremos el dato que nos indica dónde hemos de efectuar la copia de la instrucción, que en este caso es la posición 2Ø6 (2Ø6 = 2Ø1 + 5). En el segundo listado se ve el resultado final.

Direccionamiento inmediato

El modo inmediato se puede ver en la figura 5. En ella se observa que la ejecución de la instrucción suma el número 5 al dato de la posición anterior.

Se identifica fácilmente porque el dato antepone el signo #. En este caso no se trata de una dirección sino de un valor que debe ser tomado tal cual es.

```
201 DAT   5      201=203-2  |  201 DAT   5
202 SUM   3  2              |  202 SUM   3  2
203 MOV  -1  @-2            |  203 MOV  -1  @-2
... ... ... ..              |  ... ... ... ..
... ... ... ..              |  ... ... ... ..
206 ... ... ..              |  206 SUM   3  2  206=201+5
```

Figura 4. Direccionamiento indirecto.

Estructura de una instrucción

En cuanto a la estructura de cada instrucción, hay que decir que está compuesta de tres partes. En la primera está el código de la instrucción y en las otras dos se encuentra el valor de cada uno de los operandos. Cada instrucción ocupa cinco bytes en total; uno para el código y dos para cada uno de los operandos. En el byte destinado al código hay también cuatro bits que se ocupan de indicar cual es el modo de direccionamiento de cada operando (dos bits para cada uno de ellos). La figura 6 explica esto de manera gráfica.

Los códigos para los modos de direccionamientos son Ø, 1 y 2: directo, in-

```
206 DAT   7     |  206 DAT  12
207 SUM  -1 #5  |  207 SUM  -1 #5
```

Figura 5. Modo inmediato.

mediato e indirecto respectivamente.

Y ha llegado el momento de empezar a explicar cada una de las instrucciones de este lenguaje.

El repertorio completo de instrucciones con su código y una breve explicación de su función puede verse en la tabla 1 (MICROHOBBY n.º 70).

Instrucciones del lenguaje BATCODE

DAT X.

Indica que esta posición está reservada para almacenar el dato X. No es una instrucción ejecutable: si un programa intenta hacerlo en algún momento será declarado inmediatamente perdedor. Esto puede ser útil para intentar inutilizar al programa contrario. Si, por ejemplo, un programa coloca esta instrucción dentro del programa enemigo y consigue que éste trate de ejecutarla, entonces sería declarado vencedor.

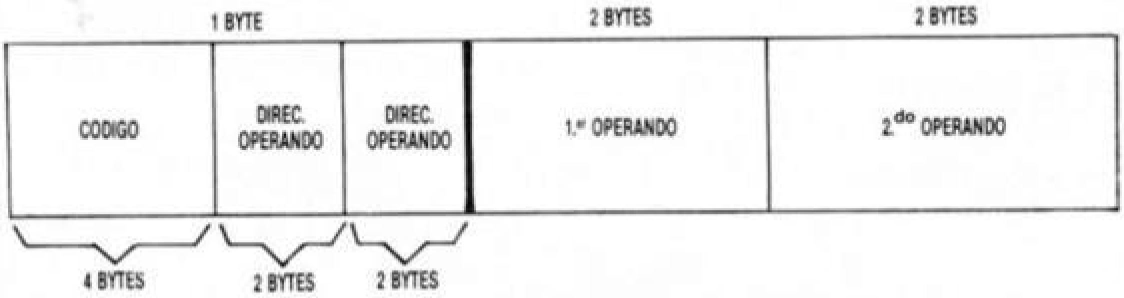


Figura 6. Formato de la instrucción.

SAL X.

Esta es una instrucción de salto, es decir, indicamos que la siguiente instrucción a ejecutar no es la que se encuentra a continuación sino la situada a X posiciones de ésta.

MOV X Y.

Nos permite mover instrucciones del lugar X al Y. Si X está indicado en modo inmediato (por ejemplo #1) colocamos la instrucción de código 1, en este caso SAL, en la posición Y. Con esta instrucción podemos bombadear las posiciones enemigas con #Ø, es decir, con instrucciones DAT pues como se ve en la tabla 1 el código de una instrucción DAT es el cero. De esta forma es posible intentar estropear el programa adversario.

También es posible, mandando los códigos adecuados, colocar nuevas instrucciones en cualquier sitio de la memoria. Hay que tener en cuenta que si por ejemplo queremos colocar la instrucción SUM que tiene código 4, este valor debe ser multiplicado por 16 ya que el código en sí ocupa los cuatro bits más significativos del byte (ver la figura 6). Por ejemplo:

3Ø5 MOV #16 2

Haría que en la línea 3Ø6 apareciera:

3Ø6 SUM Ø Ø

SUM X Y.

Esta instrucción lo que hace es sumar al primer operando de la posición X con el valor Y.

RES X Y.

Es idéntica a la anterior pero realiza la resta en vez de la suma.

SAC X Y.

Con esta instrucción entramos en un apartado de sentencias de salto condicionales. Con SAC conseguimos un salto de la posición X si el contenido de Y es igual a cero, si no, continuaremos con la instrucción siguiente. Por ejemplo:

15Ø DAT 5

151 SAL 1Ø —1

Si se ejecutara la instrucción 151 veríamos que se sigue con la línea 152. Pero si la instrucción 15Ø fuera:

15Ø DAT Ø

... entonces la ejecución saltaría a la línea 161 (161 = 151 + 1Ø).

SMY X Y.

Muy parecida a la anterior, y esta vez la pregunta se hace si el contenido de Y es mayor que cero, es decir si Y es positivo.

CMP X Y.

Esta es la instrucción general para la comparación. Aquí comparamos los valores de X e Y y saltaremos a la instrucción siguiente si X > Y; iremos dos instrucciones adelante si X = Y y saltaremos tres posiciones si X < Y.

DSC X Y.

Es una instrucción de decremento y comparación. Primero se resta de la posición Y el valor unidad y luego se salta a la línea X si el resultado es cero, en caso contrario se ejecuta la instrucción siguiente. Veamos un ejemplo:

1ØØ DAT 51

1Ø1 DSC 2 —1

1Ø2 SAL —2

En la línea 1Ø1 se resta 1 de la posición anterior y se comprueba que no es cero. El valor del dato que está en la línea 1ØØ es ahora 5Ø. Se continúa a la línea siguiente y ésta nos hace volver a la instrucción de decremento. El ciclo se parará cuando el dato valga 1, ya que entonces, al efectuar el decremento éste valdrá cero y se realizará un salto dos posiciones más adelante, prosiguiendo con el programa.

Esta instrucción es una especie de FOR...NEXT con STEP —1 y es con ella con la que realizaremos los bucles en nuestros programas.

COD X Y.

Esta y las siguientes instrucciones son las que nos permiten manipular dentro de las posiciones de memoria y tener, si se desea, un control completo sobre estos contenidos. Con COD lo que se hace es copiar el código de la instrucción indicada por X en el primer operando de la posición Y. Por ejemplo:

1 COD 5 1

2 DAT ----

......

6 SAL 11

haría que el contenido de la posición 2 fuera:

2 DAT 1

pues el código de la instrucción SAL es 1.

SET X Y.

Esta instrucción coloca el valor Y en el primer operando de la posición X. Si en el ejemplo anterior la primera línea fuera:

1 SET 1 #1

el resultado en la posición 2 será el mismo que en el caso anterior. (Fíjate que hemos utilizado # porque queríamos colocar el valor 1 y no el contenido de la posición relativa 1).

INT X.

Esta es una instrucción simple y sirve para intercambiar entre sí el contenido de los operandos de la línea X. Por ejemplo:

1 INT 9

......

1Ø SUM 4 @ 7

hará que en la posición 1Ø se intercambien los operandos:

1Ø SUM @ 7 4

LOS JUSTICIEROS DEL SOFTWARE

YABBA DABBA DOO

Muchos han sido los éxitos de la pequeña pantalla que se han visto plasmados en programas para ordenador. Si es o no una buena idea o si se atienen fielmente al original, es algo que tiene que decidir el usuario. Por lo pronto, bueno será asesorarnos sobre estos famosísimos Picapiedras, y nadie mejor para hacerlo que nuestros justicieros.

"Muy adictivo"

● *POSITIVO:*
Muy adictivo, con unos gráficos excelentes, al igual que el movimiento de los simpáticos personajes.

● *NEGATIVO:*
La falta de color, la perspectiva, que no está muy lograda y la falta de efectos sonoros.
Puntuación: 8
Fco. Javier Cano

"Lleno de detalles"

● *POSITIVO:*
Alto nivel de adicción debido a los sorprendentes gráficos y al excelente movimiento de los personajes. Un juego, además, lleno de detalles.

● *NEGATIVO:*
El sonido y la presentación de los marcadores en la pantalla.
Puntuación: 8
Pilar Arias

"Carece de música"

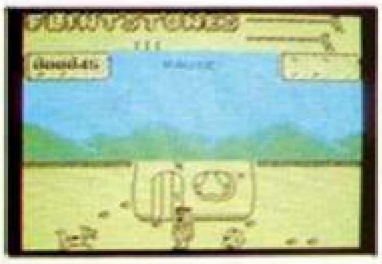
● *POSITIVO:*
Un juego muy adictivo y con unos gráficos idénticos a los de los dibujos animados.

● *NEGATIVO:*
Carece totalmente de música.
Puntuación: 8
Roberto Alonso

"Escasez de colorido"

● *POSITIVO:*
El desarrollo del juego es muy original. Los movimientos y los gráficos están muy bien realizados. Debido a que no es demasiado difícil, la adicción es elevada.

● *NEGATIVO:*
Apenas utiliza el color. Se echa de menos algo de sonido.
Puntuación: 6
Ricardo Alonso



"Alta dificultad"

● *POSITIVO:*
Los gráficos son realmente buenos. El movimiento y la presentación son más que aceptables. Es un juego bastante rápido con ideas originales y gráficos diversos. Permite seleccionar el número de vidas, con lo que la diversión está asegurada. Gran adicción y alta dificultad.

● *NEGATIVO:*
El sonido deja mucho que desear y escaso. No tiene variedad en el colorido y los gráficos se mezclan al cruzarse.
Puntuación: 8
Gabriel Martí

"Buena pantalla de presentación"

● *POSITIVO:*
La rutina de carga es fantástica, con muchos colores. La pantalla de presentación es bastante buena. Un buen detalle es poder seleccionar el número de vidas. Los movimientos están muy bien realizados mientras que los gráficos reproducen fielmente a los personajes del cómic. Se juega muy bien, tanto con el teclado como con el Joystick. Además, hay muchas pantallas y edificios diferentes.

● *NEGATIVO:*
El sonido es escaso y bastante malo. Se echa en falta un poco de colorido ya que casi todo es negro y amarillo y molesta la vista.
Puntuación: 8
David de Llodio

"Un buen programa"

● *POSITIVO:*
Es un buen programa precedido de una gran fama. Los gráficos de los personajes están muy de acuerdo con los del cómic.

● *NEGATIVO:*
Se echa en falta la variedad de colores, así como algunos efectos de sonido.
Puntuación: 7
José Antonio Galiana



MICRO Manía

Sólo para adictos

CEDIDO POR MICRO MANÍA

DRAGONTORC

Con la esperanza de que los usuarios de Spectrum puedan disfrutar aún más de este buen, pero complicadísimo juego, Juan Ignacio Velasco nos envía desde Burlada, unos cuantos consejos para dar nuevos ánimos a los amigos de Maroc. (Para la tranquilidad de todos, hemos comprobado que, a pesar del nombrecito del pueblo, esta carta no es una burla). Así pues, una vez que hayamos llegado a Locris, debemos proceder de la siguiente manera:

Primero hay que buscar tres habitaciones que tengan sus cofres respectivos. Pasando a nuestro sirviente (Hechizo Servant) por el cofre de la habitación más meridional, conseguiremos que éste se abra y podemos coger una llave que se encuentra en su interior.

Esta llave nos permitirá abrir otro cofre cercano que contiene otra llave, con la que, a su vez, podremos abrir el tercer cofre en el que, ¡Oh, maravilla!, se encuentra el hechizo LEYROD.

Ahora nos podremos dirigir con ambas llaves a la habitación de la «Flor viviente», donde si pasamos el SERVANT por un taburete, aparecerá un nuevo baúl. Este podrá ser abierto con una de las llaves que llevamos y así, la flor se transformará en un mensaje. En el interior se encuentra un RANE, del cual gustosamente nos haremos cargo.

A estas alturas la cosa estará bastante «caliente» por lo que no nos faltarán ganas de dar con el segundo RANE que se encuentra en un tesoro que previamente ha-



bremos abierto al pasar, cómo no, a nuestro SERVANT sobre él. El tercer RANE se encuentra en un esqueleto muerto, pero el cuarto, que está directamente tirado en el suelo de una de las habitaciones, no podremos cogerlo.

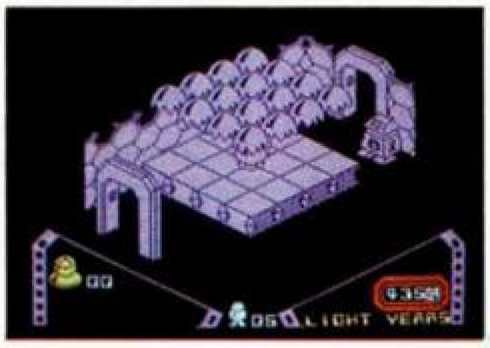
Lo que debemos hacer aquí es pasar los tres RANES que tenemos en orden diferente hasta que aquél se convierta en un cubo giratorio. Ahora no tenemos más que convertirnos en Maroc y pasar a través de él para aparecer en Wierdenge.

Allí buscaremos con el hechizo LEVROD, nuevos cubos que nos transportarán a nuevos parajes de Inglaterra.

Muchas gracias a José Ignacio y esperamos que estas pistas os sean de suma utilidad. De todas formas, aún nos queda mucho camino por recorrrer...

ALIEN 8

En números anteriores os remitimos algunos POKES muy interesantes para poder jugar con mucha más facilidad esta maravilla de la programación que responde al nombre de Alien 8. Pues ahora, Juanjo Menéndez (no sabemos si se trata del famoso actor cómico), nos envía uno nuevo para poder disfrutar de un número de vidas comprendidas entre 0 y 127.



POKE 65535, vida (vida entre 0 y 127).

Que lo disfrutéis con salud.

LIBROS



BASIC, PROGRAMACION ESTRUCTURADA Y LENGUAJE

B. Petrillo. Ed. Gustavo Gili
368 págs.

Este libro aborda en profundidad los temas relacionados con el almacenamiento de datos y los programas Basic adecuados para su gestión.

En un primer apartado hace una breve introducción a la programación, analizando su significado: el ordenador es un autómata de comportamiento riguroso, guiado por un espíritu o programa (software).

Más tarde, pasa a analizar cómo se desarrolla un programa mediante la utilización de algoritmos adecuados. Hace especial hincapié en los diagramas de flujo: a la hora de aprender, interesa más conocer la secuencia lógica, la metodología que nos lleva a resolver un determinado problema que solucionar casos concretos.

La traducción de programas en lenguaje Basic con descripción de algunos de los comandos más notables, estructuras repetitivas, el intérprete de Basic y sus instrucciones en programas estructurados (if, then, goto, etc.), y las claves de representación gráfica de algoritmos (decisiones, líneas de flujos, asignaciones, etc.), son algunos de los temas tratados en esta parte.

Posteriormente, analiza los distintos tipos de datos que existen: constantes, variables, datos numéricos y alfanuméricos.

Cómo trata el ordenador los números enteros y reales, la descomposición de un programa en módulos funcionales (subrutinas), las cadenas de caracteres sintaxis, características y funciones intrínsecas del Basic que operan sobre ellas, los datos estructurados del tipo Tabla Dimensionada y su tratamiento, ordenación y búsqueda, completan este segundo bloque.

La última parte, nos describe con profusión todo lo referente a ficheros de datos, su estructura (récord y campo), instrucciones de entrada/salida, transmisión de caracteres ASCII y de control. También comenta los sistemas de archivo de datos, haciendo alusión a los discos como soporte idóneo para ficheros de acceso aleatorio, de los que comenta sus ventajas.

CONSULTORIO

La memoria del 128K

Poseo un ordenador Spectrum 128K comprado hace unos meses y tengo la impresión de que no sea efectivamente un 128K.

*Al introducir (en modo 128K) la instrucción PRINT (PEEK 23732 + 256 * PEEK 23733-16383) /1024, para saber la capacidad, me da como resultado: 48.*

Luis LOPEZ - Tarragona

☐ Al igual que ocurre con todos los ordenadores de 128K que utilizan microprocesador Z-80, en el Spectrum 128K sólo es accesible el primer banco de memoria, es decir, 48K. Esto se debe a que el microprocesador sólo tiene 16 bits en el bus de direcciones, por lo que sólo puede direccionar 65536 posiciones de memoria.

La cuestión de cómo acceder al otro banco desde Basic es muy sencilla. La memoria supletoria está configurada como un «disco virtual» de forma que podemos almacenar en ella programas, bytes o pantallas; exactamente igual que si se tratara de un cassette sólo que la transferencia es instantánea.

Si quiere meter un programa tan largo que no le cabría en 48K, puede estructurarlo en forma de subrutinas que estarán almacenadas en el «disco-RAM» y serán recuperadas cada vez que se las necesite. Otro tanto puede hacer con las pantallas, bloques de código máquina, etc. Las instrucciones para manejar esta zona de memoria son exactamente iguales que las del cassette, pero deberán ir seguidas del signo «!» en la forma: SAVE !... LOAD !... etc.

Problemas de impresora

Me he comprado recientemente una impresora SHEIKOSA SP-1000A, y tengo algunas dudas sobre su manejo:

— No sé para qué sirve la palanca llamada «Head Position Adjustment» situada en la parte interna a mano derecha.

— He intentado utilizar, en alta calidad de letra, los caracteres definidos por el usuario pero me imprime los que trae la impresora. Me gustaría saber si esto se puede hacer y cómo.

*— El manual dice que para definir caracteres se utiliza una cuadrícula de 9 * 12, aunque la última columna no se pueda usar y tampoco la primera o la última línea, pero en la práctica, sólo puedo utilizar 7 líneas de alto en vez de 8 como indica el manual y si lo intento, me salen símbolos extraños.*

Ignacio GONZALEZ - Madrid

☐ La palanca «Head Position Adjustment» sirve para ajustar la distancia desde la cabeza de impresión al papel en función del grosor de éste. Por ejemplo, para imprimir un sobre es normal que sea necesario moverla 2 ó 3 puntos hacia atrás.

En cuanto a los caracteres definibles lo normal es que las impresoras que incluyen esta posibilidad sólo permitan definir caracteres para un tipo de letra, de forma que no podrá hacerlo, por ejemplo, para la itálica o la «NLQ» (letra de calidad).

Por otro lado, lo más frecuente es que, al definir caracteres, se puedan utilizar las 9 agujas de la impresora, bien directamente, bien haciendo que el carácter sea «descendente» mediante los atributos (caso de la Star). Aun así, debería poder usar, al menos, las 8 agujas restantes. A menos que su interface sólo sea capaz de mandar 7 bits o que la impresora acepte el octavo como bit de paridad. Es conveniente que compruebe cuantos bits de código manda su interface y configure tanto éste como la impresora para que trabajen sin paridad.

Vídeo-aventuras conversacionales

Quisiera saber los nombres de los programas de juegos de vídeo-aventuras conversacionales más famosos que existen en castellano.

Luis SELLES - Barcelona

☐ Que sepamos, sólo existen dos juegos de vídeo-aventuras conversacionales en castellano: YENGHT de Dinamic y GREMLINS de Adventure International.

Error en el número 8

En el número 8 publicasteis un esquema del interface tipo KEMPSTON en el que las resistencias iban a masa y el común de pulsadores a + 5V. Mientras que en el esquema publicado en el número 21, el común de los pulsadores va a masa y las resistencias a + 5V. ¿Cuál de estos esquemas es el correcto?

Manuel REMON - Madrid

☐ El esquema correcto del interface KEMPSTON es el publicado en el número 21. No obstante, el del número 8 funciona perfectamente si se utiliza un 74LS244 en lugar del 74LS240.

La red de área local (LAN)

¿Cómo se conectan dos Spectrum entre sí y qué utilidad tiene? ¿Hay que comprarse algún periférico para esto o bastan con los cables suministrados al comprar el ordenador? Tanto si es una cosa como otra. ¿Hay que conectarlos con el ordenador apagado?

En las fichas que regalan por la compra de su revista, deduzco que la «R» de las fichas rojas quiere decir «Rutinas»; la «I» de las azules, «Instrucciones». Pero ¿qué quieren decir la «T» y la «G» de las verdes?

Luis J. MONSALVE - Puertollano

☐ Para conectar dos Spectrum entre sí existe lo que se denomina «Red de área local» (abreviado: «LAN»). Cada Spectrum deberá estar provisto de un INTERFACE 1 y se utilizarán los cables suministrados con este interface (no los que vienen con el ordenador, esos son para el cassette). Cualquier ordenador puede ser conectado o desconectado de la «LAN» siempre y cuando la información no esté en curso.

La «T» significa «Tablas» y la «G» es «Glosario».

Listados hexadecimales

Tras utilizar ampliamente el «Cargador universal de código máquina», he adquirido un HISOFT DEVPAC

(GENS-3M y MONS-3M), a sabiendas de son los mejores ensamblador-desensamblador para el procesador Z-80.

Inicialmente, a la vista de los listados hexadecimales que publicais se me plantean los problemas siguientes:

— ¿De dónde salen todos estos caracteres?

— ¿Por qué tienen que ser 20 en cada línea?

— ¿Qué significan y de dónde salen los números de control que siguen a los 20 hexadecimales?

— ¿Qué queda exactamente en la dirección de ensamblado elegida después de entrar un texto mediante el GENS-3 y ensamblarlo sin errores?

— Cómo puede utilizarse, después de cargarla, como rutina desde el Basic?

Marcel ISACH - Barcelona

☐ En primer lugar, le recomendamos una detenida lectura de los capítulos 1 y 4 de nuestro curso de Código Máquina para aclarar los conceptos de código fuente y código objeto. Con esto, quedarían resueltas casi todas sus dudas.

Los listados que publicamos en formato adecuado al cargador universal están compuestos por el código objeto expresado en hexadecimal. Cada byte ocupa dos caracteres (desde 00 hasta FF) y cada línea tiene 10 bytes. Es por eso que hay 20 caracteres en cada línea. La razón de que sea así es, simplemente, que es el formato que necesita el cargador universal.

El número de control que está al final de cada línea es un «checksum», es decir, una suma de comprobación. Se trata, simplemente, de la suma de los 10 bytes de la línea expresada en decimal.

Tras ensamblar un programa, en la dirección indicada por «ORG» quedará el primer byte del código objeto y, a partir de ahí, todos los siguientes. Para utilizarlo desde el Basic, lo más frecuente es hacerlo mediante la función «USR», pero hay otras formas. En nuestro curso de código máquina las tratamos todas.

Ecuaciones

*Si se introduce en la memoria una fórmula, por ejemplo: A = b *h/2, ¿hay alguna manera de que el ordenador acepte los valores de «b» y «h» y devuelva el valor de «A», que acepte los valores de «A» y «b» y devuelva el valor de «h», etc.?*

Iñaki GRAO - Bilbao

☐ Un ordenador puede resolver una ecuación, lo que no puede es despejarla; queremos decir que el Basic no lo hace directamente, aunque es posible escribir un programa que lo haga.

En el caso que nos plantea, lo más fácil es tener la ecuación despejada en función de las tres incógnitas posibles y utilizar una fórmula u otra en función de cual sea la incógnita que se quiera hallar.

Manejo del cassette en C/M

¿Qué instrucciones en Assembler son las equivalentes a "LOAD... CODE y SAVE... CODE?

¿Qué se ha de hacer para que, tras cargar un bloque de código como: LOAD ""CODE 16384, 16384 podamos seguir trabajando con el ordenador sin que éste se bloquee?

Juan GARCIA - Barcelona

☐ En Assembler no existe ninguna instrucción equivalente a las que se usan en Basic para manejar el cassette. La razón es que no es una tarea que el microprocesador pueda hacer por sí mismo (tampoco hay instrucciones que multipliquen o hallen logaritmos).

No obstante, es posible manejar el cassette desde un programa escrito en Assembler, utilizando ciertas subrutinas de la ROM. En estas subrutinas habrá que entrar con el registro «IX» conteniendo la dirección de inicio del bloque, «DE» conteniendo la longitud y «A» conteniendo «00» para una cabecera o «FF» para un bloque de código. En el caso de la carga, el indicador de acarreo «C» deberá estar a «1» para cargar y a «0» para verificar.

Estas rutinas se llaman «SA-BYTES» y se encuentran respectivamente, en las direcciones: 04C2h (1218) y 0556h (1366). En una carga o verificación, se sale con el indicador de acarreo «C» a «1» si no ha habido error, y a «0» si lo ha habido.

DE OCASION

● DESEARIA que algún lector me enviase las instrucciones del juego «Southern Belle» aunque estén en inglés. Escribir a Octavio López Llamas. Sociedad, 14. 8.° B. Murcia.

● VENDO Zx Spectrum con los siguientes accesorios: teclado profesional Saga 1 Emperador, monitor fósforo verde con ampliación de sonido, impresora Seikosha GP-50S, interface Kempston y joystick, 63 revistas de Microhobby, libro de código máquina, diccionario informativo, instrucciones, interruptor de Ear y Mic, con todos sus cables, cassette especial para ordenador y sus transformadores. Todo por sólo 80.000 ptas. Interesados escribir a Fco. J. Melechón. C/ Colliure, 6, 8.° Izd. Palma de Mallorca o bien llamar al tel. (971) 29 69 96.

● DESEO ponerme en contacto con programadores de Código Máquina. Interesados escribir a Carles Jordi Fernández. C/ Carretera a Bagá, 42, 2.° Guardiola de Bergueda (Barcelona).

● CAMBIO libros de Código Máquina sobre el Spectrum. También acepto fotocopias. Interesados escribir a Fernando Vidal Sanguino. C/ Pizarro, 19. Cáceres 10003.

● CAMBIO Spectrum 48K por un Commodore CBM 64. Doy «Run», enciclopedia Práctica del Spectrum, además de revistas diversas. Lo vendo por 45.000 ptas. Llamar al tel. 773 88 55 o bien escribir a José Ignacio Paños c/ Pico de los Artilleros, 61, 2.° B. Madrid 28030.

● VENDO Interface programable con botón de reset incluido por 6.000 ptas. y joystick Tenkolet por 2.000 ptas., las dos cosas juntas por 7.000 ptas. Interesados escribir a Fco. Javier Suárez Domínguez. C/ Ctra. General, 88. Punta de Hidalgo. Tenerife. También vendo Spectrum Plus en perfecto estado, comprado en Ago-85, con todos los cables manuales, etc. Por el precio de 35.000 ptas. Si te interesa escribe a la dirección anterior.

● VENDO Zx Spectrum 48K, completamente nuevo en garantía, con su embalaje y todos los accesorios, manual en castellano, cinta Horizontes y regalo rápido. Precio 22.000 ptas. Interesados escribir a Eduardo M.S. Chimisay, bl-30, 2.° B. S/C de Tenerife. Apartado de correos 38010.

● VENDO los números 1 al 19, 43 y 48 de Microhobby y los números 5 al 11 de ZX, por 2.500 ptas. También vendo o cambio por otras revistas la enciclopedia «Electrónica fundamental» (6 libros). Regalo un libro sobre colores y gráficos del Spectrum. Interesados llamar al tel. (925) 21 36 25. José Luis.

● VENDO por cambio de modelo, ordenador Dragón 32 en perfecto estado. Comprado en Dic-85. Precio 28.000 ptas. (negociables). Interesados escribir a Pablo Mielgo del Riego. H.H. Maristas. Apto., 3. Tuy. Pontevedra o bien llamar al tel. (986) 60 02 14 (2 a 4 de 9 a 10)

● CAMBIO interface programable Indescomp por interface Kempston con joystick. Interesados escribir a Carles Jordi Fernández. C/ Ctra. a Bagá, 42, 2.°. Guardiola de Bergueré. Barcelona.

● BUSCO usuarios, de Madrid, con impresora y Spectrum para que me listen programas y archivos. Pago los trabajos y el papel. Interesados escribir a M. Alvarez. Castellana, 270, 4.° B. Madrid 28016.

● COMPRARIA fotocopias del juego conversacional Spiderman y Spy Hunter, Jump Challen, Shadow Fire o bien cambiaría por otras. Interesados escribir a la siguiente dirección: José Luis Fernández Jambrina. C/ Hernán Cortés, 32, 2.° A. 49003 Zamora.

● VENDO lápiz óptico DK'Tronics comprado el 20-1-86, al precio de 3.250 ptas. En perfecto estado. Contactar con Adolfo García (hijo) al tel. (91) 441 50 84 (7 a 10).

● VENDO Atari vídeo-computer System, con instrucciones en castellano por sólo 40.000 ptas. Contactar con Gabriel. Gran Vía, 146, 2.° A. Vigo. Pontevedra. Tel. (986) 42 23 67.

● VENDO Zx Spectrum 16K, comprado en dic-85 (sin usar) con alimentador, cables, cinta de introducción al Basic (horizontes) manual en castellano y garantía Investrónica. Precio: 18.000 ptas. Interesados llamar al (91) 245 97 84 y preguntar por Arturo Dahlander Varela (noches).

● CLUB Electronic Hard. Un club de usuarios del Spectrum 48K, 64K y 128K. Anímate y escribe, para pedir más información a la siguiente dirección: Pedro. M.R. c/ Once de Noviembre, 2. Puertollano. Ciudad Real.

● COMPRO tu ordenador sea cual sea su marca. Sólo si está en buenas condiciones de funcionamiento. Te doy hasta 16.000 ptas. Interesados escribir a la siguiente dirección: José Manuel Español c/ San Cristóbal, 73, 2.°, 3.°. Constanti. Tarragona.

● VENDO un vídeo-juego Philips G-7000 en perfecto estado. Comprado en mar-85. Incluidos dos mandos y cartuchos. Todo por el precio de 15.000 ptas. Interesados llamar al tel. 46 66 45. Murcia. Preguntar por José Luis.

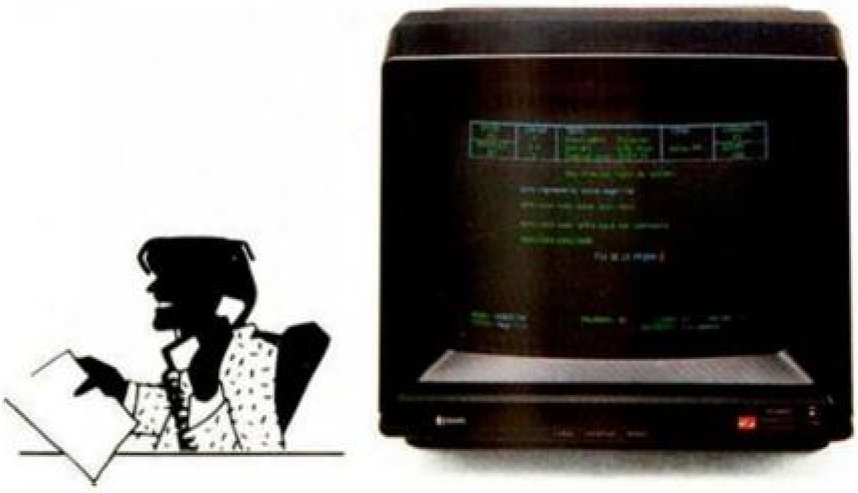
● DESEARIA contactar con usuarios del Spectrum. Interesados contactar con Javier Mañas. C/ Sevilla, 21, 5.° C. Zaragoza o bien llamar al tel. (976) 37 12 56.

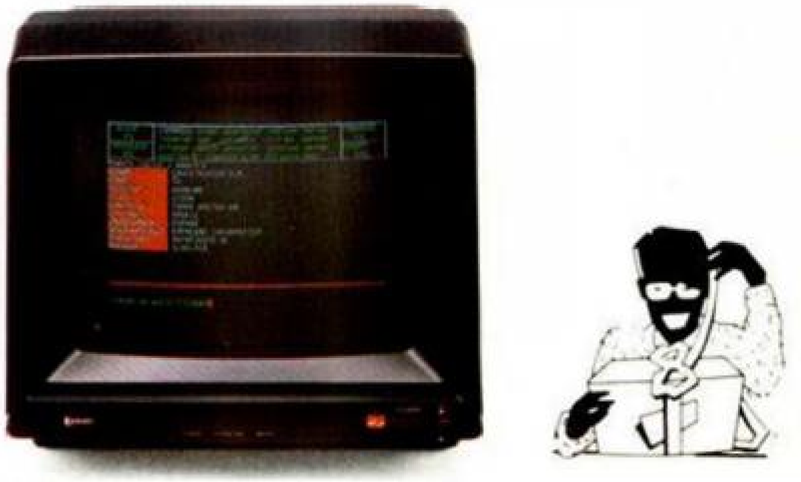
● CAMBIO monitor GT-65 de fósforo verde por monitor a color pagando diferencia a convenir. Interesados llamar al tel. (93) 331 13 95. Valencia. (Sólo tardes).

● VENDO órgano Botempi, con pequeño mueble incorporado. Además incluyo libros, y partituras. O bien lo cambiaría por un ordenador. Interesados dirigirse a Ricardo Pérez Font C/ Bruselas, 41. Madrid 28028. Tfno: (91) 245 08 34.

● VENDO Zx Spectrum 48K, en buen estado, completo y con salida monitor además para TV, también incluyo revistas, interface para joystick tipo Kempston y otro programable, reset, cassette especial para ordenador marca Gold King, todo por 30.000 ptas. Preguntar por Carlos al tel. (91) 463 15 36 (sólo tardes).

ASI TRABAJA EL QL HECHO PARA NOSOTROS





QL QUILL. TRATAMIENTO DE TEXTOS

QL Quill muestra en pantalla exactamente cómo aparecerá su escrito cuando esté impreso. Olvídese de pesadas memorizaciones de comandos. Con QL Quill tiene siempre todo lo necesario en pantalla. Escriba en negrita, corte, una, fije tabuladores, márgenes, sobreescriba... después de todo esto su QL le dirá, además, cuántas palabras ha escrito.

QL ARCHIVE. BASE DE DATOS

Organizar su agenda o poner al día su fichero se convierte en un trabajo agradable. QL Archive resuelve rápidamente su problema con un sistema de archivo de gran facilidad de uso, usando un lenguaje aún más sencillo que el BASIC.

Con un mínimo de práctica puede crear desde un simple directorio a una extensa base de datos. Podrá determinar relaciones, extraer datos, reorganizarlos, hacer cálculos estadísticos y contables.

Con QL Archive podrá crear un formato personalizado para sus informes.





QL ABACUS. HOJA ELECTRONICA DE CALCULO

Este programa ha superado con mucho las tradicionales hojas de cálculo de otros ordenadores. Con QL Abacus escriba simplemente "costes de personal", o "personal", o "pers" para encontrar esa celdilla. QL Abacus también le ayuda a decidir lo que hay que hacer mediante diferentes sugerencias, dentro de una amplia gama de posibilidades. Podrá, además, crear hojas electrónicas de cálculo con más de 6.000 celdas individuales.

Podrá mostrar ventanas múltiples, variar la distancia entre columnas...

QL EASEL. GRAFICOS

Olvídese de construir tablas de valores o enfrentarse con situaciones dudosas antes de ver algún resultado. Con QL EASEL podrá desde el principio crear gráficos. Inmediatamente creará diagramas de barras, filas, líneas rellenas, sectores, barras sobrepuestas...

Todo ello simplemente pulsando una tecla.

Introduzca textos donde quiera. Dentro o cerca de su diagrama. Más aún, mueva el texto por la pantalla para ver dónde queda mejor... en el color que prefiera.

DISTRIBUIDOR EXCLUSIVO

investronica

Tomás Bretón, 60. Telf. (91) 467 82 10. Telex 23399 IYCO E. 28045 Madrid
Camp, 80. Telf. (93) 211 26 58-211 27 54. 08022 Barcelona